[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/445 (2006.01)

[12] 发明专利申请公开说明书

[21] 申请号 200610071181.2

[43] 公开日 2006 年 9 月 13 日

[11] 公开号 CN 1831774A

[22] 申请日 2006.2.5
[21] 申请号 200610071181.2
[30] 优先权
[32] 2005. 2. 2 [33] US [31] 60/649486
[71] 申请人 印西德软件公司
地址 台湾省台北市
[72] 发明人 R·A·弗林

[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 张志醒

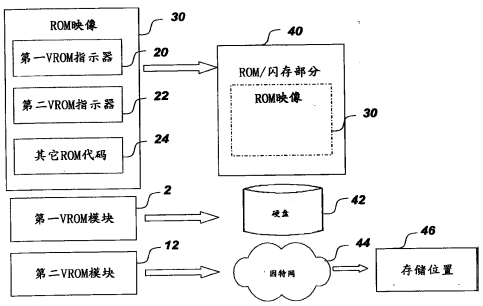
权利要求书 8 页 说明书 18 页 附图 17 页

[54] 发明名称

用于减小固件的存储需求并提供固件的安全更新和存储区域的系统和方法

[57] 摘要

讨论了一种使增加数量的固件对于计算机预启动可用的机制。为了增加对预启动可用的固件数量，设计决定在建立过程期间哪些固件的段需要放置在 ROM 部分以及哪些固件的段可以放置在其它地方。远离 ROM 存储的固件的段称作“虚拟 ROM 模块”。虚拟 ROM 模块的每一个被指定一个产生的唯一标识符，使用算法如 MD5 或 SHA－1 为每个模块构造“消息摘要”。在 ROM 映像的软件建立中，为每个虚拟 ROM 模块创建的消息摘要—唯一标识符对用作对虚拟模块的逻辑指示器。另外，把搜索路径变量设置到非易失性存储器的 ROM 映像中。该搜索路径为搜寻虚拟 ROM 模块提供一个或多个位置，并可以在随后时间点被更新。本发明也允许安全存储和固件的更新。



1、一种用于减少存储在电子装置中的固件的存储需求的方法，包括：

划分固件映像以创建多个虚拟 ROM 模块，所述多个虚拟 ROM 模块存储在位于所述电子装置的 ROM 映像外部的至少一个位置中；

包括在固件建立时设置在所述 ROM 映像中的对于所述多个虚拟 ROM 模块的每一个的参考，每一个参考包括用于相关虚拟 ROM 模块的消息摘要和唯一标识符；

在预启动期间使用与所述多个虚拟 ROM 模块中所选择的一个相关的所述参考中的所述唯一标识符来检索所述多个虚拟 ROM 模块中所选择的一个，以及

在预启动期间使用与所述多个虚拟 ROM 模块中所选择的一个相关的所述参考中的所述消息摘要来验证所述多个虚拟 ROM 模块中所选择的一个。

2、如权利要求 1 所述的方法，进一步包括：

把搜索变量设置到 ROM 映像中，所述搜索变量指示从中可能检索所述多个虚拟 ROM 模块其中之一的至少一个位置。

3、如权利要求 2 所述的方法，进一步包括：

使用设置在所述 ROM 映像中的所述搜索变量检索所述多个虚拟 ROM 模块其中之一。

4、如权利要求 1 所述的方法，其中所述 ROM 映像存储在 ROM(只读存储器)、PROM(可编程 ROM)、EPROM(可擦除 PROM)、EEPROM(电可擦除 PROM)和闪存的组中的其中一个中。

5、如权利要求 1 所述的方法，进一步包括：

在多于一个位置中存储所述多个虚拟 ROM 模块。

6、如权利要求 5 所述的方法，其中所述多个虚拟 ROM 模块中的至少一个存储在通过网络可访问的位置中。

7、如权利要求 1 所述的方法，其中所述多个虚拟 ROM 模块中的至少一个是 EFI 架构固件卷。

8、如权利要求 1 所述的方法，其中存储在所述 ROM 映像中的至少一个固件文件存储在 EFI 固件卷中。

9、一种用于减少存储在电子装置中的固件的存储需求的方法，包括：

划分固件映像以创建虚拟 ROM 模块，所述虚拟 ROM 模块存储在位于所述电子装置的 ROM 映像外部的位置中；

包括在固件建立时设置在所述 ROM 映像中的对于所述虚拟 ROM 模块的参考，所述参考包括用于所述虚拟 ROM 模块的消息摘要和唯一标识符；

在预启动期间使用与所述虚拟 ROM 模块相关的所述参考中的所述唯一标识符来检索所述虚拟 ROM 模块，以及

在预启动期间使用与所述虚拟 ROM 模块相关的所述参考中的所述消息摘要验证所述虚拟 ROM 模块的真实性。

10、一种用于减小存储在电子装置中的固件的存储需求的系统，包括：

在建立过程中创建的多个虚拟固件映像模块，所述多个虚拟 ROM 模块存储在位于所述电子装置的固件 ROM 映像外部的至少一个位置中；和

固件 ROM 映像，所述固件 ROM 映像包括对于所述多个虚拟 ROM 模块的每一个的参考，每一个参考包括用于相关的虚拟 ROM 模块的消息摘要和唯一标识符，所述唯一标识符用于在预启动期间检索所参考的映像模块，所述消息摘要用于在执行所检索的映像模块之前验证所述映像模块的真实性。

11、如权利要求 10 所述的系统，其中所述 ROM 映像存储在 ROM(只读存储器)、PROM(可编程 ROM)、EPROM(可擦除 PROM)、EEPROM(电可擦除 PROM)和闪存的组中的其中一个中。

12、如权利要求 10 所述的系统，其中所述多个虚拟 ROM 模块的至少一个是 EFI 架构固件卷。

13、如权利要求 10 所述的方法，其中存储在所述 ROM 映像中的至少一个固件存储在 EFI 固件卷中。

14、一种保存用于减小在电子装置中的固件的存储需求的计算机可执行指令的媒介，所述指令包括：

划分固件映像以创建多个虚拟 ROM 模块的指令，所述多个虚拟 ROM 模块存储在位于所述电子装置的 ROM 映像外部的至少一个位置中；

包括在固件建立时设置在所述 ROM 映像中的对于所述多个虚拟 ROM 模块的每一个的参考的指令，每一个参考包括用于相关虚拟 ROM 模块的消

息摘要和唯一标识符；在预启动期间使用与所述多个虚拟 ROM 模块中所选择的一个相关的所述参考中的所述唯一标识符来检索所述多个虚拟 ROM 模块中所选择的一个的指令，以及

在预启动期间使用与所述多个虚拟 ROM 模块中所选择的一个相关的所述参考中的所述消息摘要来验证所述多个虚拟 ROM 模块中所选择的一个的指令。

15、一种安全地更新电子装置中的固件的方法，包括：

划分固件映像以创建多个虚拟映像模块，所述多个虚拟映像模块存储在位于所述电子装置的 ROM 映像外部的至少一个位置中；

提供用于在预启动期间验证更新的虚拟映像模块的更新验证过程，所述更新验证过程由唯一标识符识别并包括解密处理；

包括在固件建立时设置在所述 ROM 映像中的对于所述多个虚拟映像模块的每一个的参考，每一个参考包括用于相关的虚拟映像模块的版本标识符、消息摘要和唯一标识符，在所述 ROM 映像中固件的建立也包括对所述更新验证过程的参考；

提供包括版本标识符的更新的虚拟映像模块；和

在预启动期间使用由所述 ROM 映像中的参考识别的所述更新验证过程来验证所述更新的虚拟映像模块。

16、如权利要求 15 所述的方法，进一步包括：

利用所述更新的虚拟映像模块代替在所述 ROM 映像中参考的所述多个虚拟映像模块中的一个。

17、如权利要求 15 所述的方法，进一步包括：

产生用于所述更新的虚拟映像模块的消息摘要；

通过加密处理对所述更新的虚拟映像模块的消息摘要进行加密，所述加密处理需要包含在所述更新验证过程中的解密处理来对所加密的消息摘要进行解密。

18、如权利要求 17 所述的方法，进一步包括：

使用包含在所述更新验证过程中的所述解密处理对所加密的消息摘要进行解密，所述解密作为所述更新的虚拟映像模块的验证的一部分执行。

19、如权利要求 18 所述的方法，进一步包括：

将所加密的消息摘要的解密与所述更新验证过程产生的新消息摘要比较以验证所述更新的虚拟映像模块。

20、如权利要求 15 所述的方法，进一步包括：

为所述更新的虚拟映像模块产生消息摘要；

用私钥对所述消息摘要进行加密；和

利用与所述私钥相关的公钥对所加密的消息摘要进行解密，由所述更新验证过程使用所述公钥。

21、如权利要求 15 所述的方法，其中所述更新验证过程将所述多个虚拟映像模块的其中之一的版本标识符与所述更新的虚拟映像模块的版本标识符比较以将所述更新的虚拟映像模块识别为新模块。

22、如权利要求 15 所述的方法，其中所述更新验证过程作为所述 ROM 映像的一部分被存储。

23、如权利要求 15 所述的方法，其中所述更新验证过程作为所述多个虚拟映像模块的其中之一存储。

24、如权利要求 15 所述的方法，其中所述更新的虚拟映像模块是 EFI 架构固件卷。

25、如权利要求 15 所述的方法，进一步包括：

为所述更新的虚拟映像模块产生消息摘要；

通过对称密钥对所述消息摘要进行加密；和

利用所述对称密钥对所加密的消息摘要进行解密，所述对称密钥由所述更新验证过程使用。

26、一种安全地更新电子装置中的固件的方法，包括：

划分固件映像以创建虚拟映像模块，所述虚拟映像模块存储在位于所述电子装置的 ROM 映像外部的至少一个位置中；

提供用于在预启动期间验证更新的虚拟映像模块的更新验证过程，所述更新验证过程由唯一标识符识别并包括解密处理；

包括在固件建立时设置在所述 ROM 映像中的对于所述多个虚拟映像模块的参考，所述参考包括用于相关的虚拟映像模块的版本标识符、消息摘要和唯一标识符，在所述 ROM 映像中固件的建立也包括对所述更新验证过程的参考；

提供包括版本标识符的更新的虚拟映像模块；和

在预启动期间使用由所述 ROM 映像中的所述参考识别的所述更新验证过程来验证所述更新的虚拟映像模块。

27、一种安全地更新电子装置中的固件的方法，包括：

划分固件映像以创建多个虚拟映像模块，所述多个虚拟映像模块存储在位于所述电子装置的ROM映像外部的至少一个位置中；

提供用于在预启动期间验证更新的虚拟映像模块的更新验证过程，所述更新验证过程由唯一标识符识别并包括解密处理；

包括在固件建立时设置在所述 ROM 映像中的对于所述多个虚拟映像模块的每一个的参考，在所述 ROM 映像中固件的建立也包括对所述更新验证过程的参考；

在可信任服务器上提供更新的虚拟映象模块；

使用所述 ROM 映象中的所述参考所识别的所述更新验证过程在所述电子装置和所述可信任服务器之间产生安全通道；以及

使用所述通道下载已识别的更新虚拟映像模块到所述电子装置中，该下载的虚拟映像代替在所述 ROM 映像中参考的所述多个虚拟映像模块中的一个。

28、一种安全地更新在电子器件中固件的系统，包括：

在建立过程中创建的多个虚拟固件映像模块，所述多个虚拟 ROM 模块存储在位于电子装置的固件ROM映像外部的至少一个位置中；

包括在验证更新的虚拟模块时使用的解密处理的更新验证过程；和

固件 ROM 映像，所述固件 ROM 映像包括对所述更新验证过程的参考和对于所述多个虚拟 ROM 模块的每一个的参考，每一个参考包括用于相关的虚拟 ROM 模块的版本标识符、消息摘要和唯一标识符，所述唯一标识符用于在预启动期间检索所参考的虚拟映像模块，所述消息摘要用于在执行所检索的映像模块之前验证所述虚拟映像模块的真实性。

29、如权利要求 28 所述的系统，其中所述更新的虚拟映像模块是 EFI 架构固件卷。

30、一种保存用于安全地更新电子装置中的固件的计算机可执行指令的媒介，该指令包括：

划分固件映像以创建多个虚拟映像模块的指令，所述多个虚拟映像模块存储在位于所述电子装置的ROM映像外部的至少一个位置中；

提供用于在预启动期间验证更新的虚拟映像模块的更新验证过程的指令，所述更新验证过程由唯一标识符识别并包括解密程序；

包括在固件建立时设置在所述 ROM 映像中的对所述多个虚拟映像模块的每一个的参考的指令，每一个参考包括用于相关虚拟映像模块的版本标识符、消息摘要和唯一标识符，在所述 ROM 映像中固件的建立也包括对所述更新验证过程的参考；

用于提供包括版本标识符的更新的虚拟映像模块的指令；和

在预启动期间使用由所述 ROM 映像中的所述参考识别的所述更新验证过程来验证所述更新的虚拟映像模块的指令。

31、一种用于提供装载操作系统前从 ROM 执行的计算装置的固件的安全存储的方法，包括：

提供保存固件的指定安全存储容器，所述固件在装载所述计算装置的所述操作系统之前执行；

使用安全存储加密密钥对所述指定安全存储容器进行加密；

包括在固件建立时设置在ROM映像中的对所加密的存储容器的参考；

使用至少一个唯一标识符将所加密的安全存储容器写入所述 ROM 外部的第二存储；

在装载所述计算装置的所述操作系统之前使用所述 ROM 映像中的所述参考检索所加密的安全存储容器。

32、如权利要求31所述的方法，进一步包括：

对所加密的指定安全存储容器产生消息摘要；和

在所述参考中包括所述消息摘要和至少一个唯一标识符。

33、如权利要求32所述的方法，进一步包括：

对所检索的加密安全存储容器产生新消息摘要；

通过比较消息摘要来识别所述新消息摘要和所述参考中的消息摘要之间的匹配；和

使用所述安全存储加密密钥对所述安全存储容器进行解密。

34、如权利要求 31 所述的方法，其中所述安全存储加密密钥存储在所

述 ROM 映像的一部分中，在装载所述计算装置的所述操作系统之前所述 ROM 映像被装载到不可读存储单元。

35、如权利要求 31 所述的方法，其中所述至少一个唯一标识符包括用于所述计算装置的唯一机器标识符。

36、如权利要求 35 所述的方法，其中所述唯一机器标识符存储在所述 ROM 映像的一部分中，在装载所述计算装置的所述操作系统之前所述 ROM 映像被装载到不可写存储单元。

37、如权利要求 31 所述的方法，进一步包括：

通过网络访问所述第二存储。

38、如权利要求 31 所述的方法，其中所述指定安全存储容器是 EFI 架构固件卷。

39、如权利要求 31 所述的方法，进一步包括：

提供第二指定安全存储容器；

产生唯一公-私密钥对；

在所述指定安全存储容器中存储所述唯一公-私密钥对的公钥，在所述第二指定安全存储容器中存储所述唯一公-私密钥对的私钥，存储公钥的所述指定安全存储容器被装载到标为不可写的存储器中，所述第二指定安全存储容器被装载到标为不可读的存储器中；

利用预启动固件来标记创建用于递送到其它环境的内容，所述标记包括利用所述唯一公-私密钥对的所述私钥对内容的加密，所述内容在标记后递送；和

使用所述公-私密钥对的所述公钥对所标记的内容进行解密。

40、如权利要求 39 所述的方法，其中所述标记过程在利用所述私钥标记所述内容之前对所述内容运行消息摘要。

41、如权利要求 39 所述的方法，其中在利用所述公-私密钥的所述公钥对所述内容解密之前重新产生消息摘要以便验证所标记的内容。

42、一种用于提供从 ROM 执行预启动的计算装置的固件的安全存储的方法，包括

固件 ROM 映像，所述固件 ROM 映像包括对至少一个加密的指定安全存储容器的参考，所述加密的指定安全存储容器保存装载所述操作系统前要

执行的固件，对所述至少一个加密的指定安全存储容器的参考包括用于所述加密的指定安全存储容器的消息摘要，使用所述消息摘要以便在执行所述加密的指定安全存储容器中的固件之前验证所述加密的指定安全存储容器的真实性；

指定的安全存储容器，所述指定的安全存储容器由安全存储加密密钥加密并存储在与所述 ROM 分开的第二存储中；和

安全存储加密密钥，用于最初对所述指定的安全存储容器加密并在从所述第二存储检索后对所述指定的安全存储容器进行解密，在装载所述操作系统之前所述安全存储容器位于标记为不可读的所述 ROM 映像的一部分中。

43、如权利要求 42 所述的系统，其中所述固件 ROM 映像最初存储在 ROM(只读存储器)、PROM(可编程 ROM)、EPROM(可擦除 PROM)、EEPROM(电可擦除 PROM)和闪存的组中的其中之一中。

44、如权利要求 42 所述的系统，其中所述固件 ROM 映像包括对第二指定的加密安全存储区的第二参考。

45、一种保存计算机可执行指令的媒介，所述指令提供装载操作系统前从 ROM 执行的计算装置的固件的安全存储，所述指令包括：

提供保存固件的指定安全存储容器的指令，所述固件在装载所述计算装置的所述操作系统之前执行；

使用安全存储加密密钥对所述指定安全存储容器进行加密的指令；

包括在固件建立时设置在 ROM 映像中的对所加密的存储容器的参考的指令；

使用至少一个唯一标识符将所加密的安全存储容器写入所述 ROM 外部的第二存储的指令；

在装载所述计算装置的所述操作系统之前使用所述 ROM 映像中的所述参考检索所加密的安全存储容器的指令。

用于减小固件的存储需求并提供固件的安全更新和存储区域的系统和方法

技术领域

一般来说，本发明的说明性实施例涉及在 PC 上执行 POST 的固件、包括 BIOS 固件，更具体地说，涉及减小固件的存储需求、安全地存储固件并以安全方式更新固件。

相关申请

本发明要求2005年2月2日提交的美国申请号为60/649486标题为"*System and Method for reducing Merory Requirements of Firmware and Providing Secure Updates and Storage Areas for Firmware*"的美国临时申请的权益和优先权。

背景技术

近年来，对 PC 预启动所利用的大量固件代码的需求增大。固件是已经写入到只读存储器(ROM)模块的软件，该模块包括但不局限于 ROM、PROM、EPROM、EEPROM 和闪存。对固件需求增加的一个原因是在启动 PC 和准备载入操作系统(OS)时需要通过固件(BIOS 或架构)(这里使用的术语"架构"指 Intel 公司(Santa Clara，California)的"EFI 平台创新架构"，并在下面将进一步讨论)进行操作的复杂度增加。需求增加的另一原因是在操作系统装载前创建在预启动环境中运行的额外增值特征在产业中存在相当大的利益。

不幸的是，存在许多与使用固件的传统方法相关的问题。由于产业中存在将 PC 的硬件成本降低的强刺激的事实，在 ROM 或闪存部件上可利用的存储的增加已不能完全满足对预启动可利用的更大量固件代码增加的需求。另外，增加固件代码的需求也需要与其它软件环境通信的安全装置、包括因特网上的服务器和在操作系统装载后运行的代码。并且，虽然存在用于安全内容预启动的解决方法，但是这些现有的解决方法依赖于在本地硬盘上可用的专用分区，或者依赖于能够存储和隐藏密钥的单独的安全芯片。对于以安全方式更新在闪存部件存储的固件的任务，也引起了广泛的关注。由于不能顺利完成更新会导

致计算机变得不能工作，所以更新存储在闪存部件上的固件的现有方法包括对计算机的未来可操作性的风险。允许这样的更新也会产生小的安全隐患，因为闪存固件经常会完全访问计算机的内部。

发明内容

本发明的示例实施例提供了一种用于使更多数量固件对于计算机预启动可用的机制。为了增加预启动可用的固件数量，在建立过程期间设计决定固件的哪些段需要设置在ROM部分并且固件的哪些段可以位于其它地方。远离ROM存储的固件段称为“虚拟ROM模块”。为每个虚拟ROM模块指定一个产生的唯一标识符，使用算法如MD5或SHA-1为每个模块创建“消息摘要”。消息摘要唯一地代表虚拟模块，使得如果虚拟模块改变时，消息摘要也改变。在ROM映像(image)的软件创建中，为每个虚拟ROM模块创建的消息摘要-唯一标识符对用作虚拟模块的逻辑指示器(pointer)。另外，把搜索路径变量设置到非易失性存储中的ROM映像中。该搜索路径提供寻找虚拟ROM模块的一个或多个位置，并可以在后面的时间点被更新。

在制造过程期间把实际ROM映像(包含指向虚拟模块的逻辑指示器)设置到PC上的ROM或闪存部分。当实际ROM映像要执行虚拟ROM模块时，它检索与虚拟ROM模块相关的唯一标识符，并使用搜索路径变量和唯一标识符定位虚拟ROM模块的副本。它可以从本地磁盘、从CD ROM或任何其他存储媒介装载模块，以及通过网络下载模块。一旦已经装载了模块，它在模块上运行相同的消息摘要算法，并在执行前确认该模块没有改变。

本发明的示例实施例也允许以安全方式更新固件。在实际ROM中使用两个附加属性以涉及虚拟ROM模块。这两个附加属性是版本属性和对能验证更新的单独模块的参考。这个单独模块可以或不可以并入实际ROM中一它可以是用于另一虚拟ROM模块的唯一标识符。更新过程更新与第一虚拟ROM模块相关的消息摘要和与第一虚拟ROM模块相关的版本属性。该更新过程也产生了相应文件的新副本(其可以位于本地磁盘)，当以对应于上述过程的方式“散列”相应文件时(“散列”在这种情况下意思是对文件运行消息摘要算法中其中一个)会“匹配”新消息摘要。

在下载更新模块后，验证模块检验进行更新是可接受的。通过已知用于验

证经过因特网传播的内容的多个已知机制的其中之一如公-私密钥对来验证经过因特网传播的内容，以便使用 SHA-1 算法加密或解密由内容制成的消息摘要。可以使用多个更新验证模块，每一个更新验证模块是指由不同的固件供应商提供的固件，每一个该供应商可以采用不同的确认配置。即使在不同的固件供应商使用相同的固件机制的情况下，每个供应商可以使用他们自己的用于标记内容的公-私密钥对。

本发明的示例实施例也提供个人计算机的安全存储的创建和访问安全存储。为了安全存储，ROM 或闪存部件的区域需要被做成对预启动不运行的随后代码要么只读要么完全不可读。本发明使 ROM 或闪存部件的区域只读(称作“不可写入”)和完全不可读(指“不可读”)。为了检测存储器上的改动需要不可写入，而为了能够对私钥加密需要“不可读”。

创建和访问安全存储区域的过程在实际 ROM 中存储“虚拟 ROM”模块参考或指示器，该参考或指示器包括待检索的模块的唯一标识符，实际 ROM 包含用于整个机器的产生的唯一标识符。在检索虚拟 ROM 模块时，同时使用模块标识符和机器标识符。一旦检索到虚拟ROM模块，使用存储在虚拟ROM模块参考中的消息摘要验证模块。如果需要，在此之后，使用存储在实际ROM别处的密钥来对虚拟ROM模块解密。

预启动码在存储器中对虚拟 ROM 模块(其实质上是安全存储单元)更新。在完成这些更新的时间点，虚拟 ROM 模块被写回到其被检索的位置。如果需要，通过第一次使用密钥对模块加密、产生虚拟 ROM 模块的新消息摘要、并使用模块标识符和机器标识符将新模块写回到其存储位置，写出涉及颠倒上述步骤。同样，在实际 ROM 中的虚拟 ROM 模块参考必须被更新以反映新的消息摘要值。随后，在实际 ROM 中的虚拟 ROM 模块参考和用于 PC 的机器标识符是不可写入的。另外，如果该存储被加密，并且使用密钥，则包含密钥的实际ROM区是不可读的。

在本发明的一个方面，用于减小存储在电子装置中的固件的存储要求的方法包括划分固件映像以创建多个虚拟 ROM 模块的步骤。多个虚拟 ROM 模块存储在位于电子装置上的 ROM 映像外部的至少一个位置处。该方法还包括在固件建立时设置在 ROM 映像中的对于多个虚拟 ROM 模块的每一个的参考。每个参考包括用于相关虚拟 ROM 模块的消息摘要和唯一标识符。该方法也在

预启动期间使用与多个虚拟 ROM 模块中所选择的一个相关的参考中的唯一标识符来检索多个虚拟 ROM 模块中所选择的一个。该方法在预启动期间使用在与多个虚拟 ROM 模块中所选择的一个相关的参考中的消息摘要进一步验证从多个虚拟ROM模块中所选择的一个。

在本发明的另一方面，用于减小存储在电子装置中的固件的存储要求的系统包括在建立过程中创建的多个虚拟固件映像模块。该多个虚拟 ROM 模块存储在位于电子装置上的固件 ROM 映像外部的至少一个位置上。该系统也包括固件 ROM 映像。该固件 ROM 映像包括对于多个虚拟 ROM 模块每一个的参考，每一个参考包括相关虚拟 ROM 模块的消息摘要和唯一标识符。该唯一标识符用于在预启动期间检索参考的映像模块，该消息摘要用于在执行检索的映像模块之前验证映像模块的真实性。

在本发明的一个方面，用于减小存储在电子装置中的固件的存储需求的方法包括划分固件映像以创建虚拟 ROM 模块的步骤，该虚拟 ROM 模块存储在位于电子装置的 ROM 映像外部的位置。该方法也包括在固件建立时设置在ROM映像中的对于虚拟ROM模块的参考。该参考包括用于虚拟ROM模块的消息摘要和唯一标识符。另外，该方法也在预启动期间使用与虚拟 ROM 模块相关的参考中的唯一标识符来检索虚拟 ROM 模块。在预启动期间使用在与虚拟ROM模块相关的参考中的消息摘要验证虚拟ROM模块的真实性。

附图说明

本发明在所附权利要求书中提出了其特殊性。结合附图参考下列描述，将会更好地理解上述本发明的优点和本发明的更多优点，附图中：

图 1 示出 VROM 模块和产生的消息摘要和 GUID 之间的关系；

图 2 示出在创建实际 ROM 映像时 VROM 逻辑指示器的使用；

图 3 示出分布过程的视图，其中实际 ROM 映像存储在 ROM 部件中，并且虚拟ROM模块可以存储在磁盘上或因特网上；

图 4 是建立过程的流程图；

图 5 示出检索过程的视图，其中使用虚拟 ROM“指示器”检索未驻留在实际ROM部件中的虚拟ROM模块；

图 6 是检索过程的流程图；

图 7 示出在建立过程期间为每个 VROM 模块产生的附加信息、版本号和对更新有效器的参考；

图 8 示出存储在 ROM 映像中的更新有效器和“修改”的 VROM 指示器；

图 9 示出用于更新有效器的公-私密钥的产生；

图 10 示出产生“密钥”的流程图，该“密钥”与使用公-私密钥对的私钥的模块更新相关联；

图 11 示出更新的 VROM 模块的检索和确认；

图 12 示出用于更新的检索的流程图，包括对验证密钥的流程图的参考；

图 13 示出用于密钥的验证的流程图；

图 14 示出用于创建固件的安全存储区的组件的分解(breakdown)；

图 15 示出对安全存储区加密和将安全存储区保存到第二存储位置(在这种情况下，硬盘)。

图 16 示出第一次和随后写入安全存储的流程图；和

图 17 示出第一次和随后读取安全存储的流程图。

具体实施方式

本发明的说明性实施例增加了可用于计算机预启动的固件数量，提供了对固件的安全更新机制，并提供了用于创建和访问安全存储区的方法。为了更清楚的解释，下面依次分别讨论本发明的每一方面。这里描述的本发明的示例实施例涉及到 API 的英特尔 EFI 平台创新架构(“架构”)。转换成非架构解决方案之上的实施例对于本领域的技术人员来说是显而易见的。另外，虽然该讨论集中于 PC，但也认为包括存储在非易失性存储部件上的预编程固件映像的其它非 PC 计算装置落入本发明的范围内。

在详细讨论本发明之前，讨论由本发明使用的 BIOS 和架构组件将是有帮助的。英特尔 EFI 平台创新架构(“架构”)是 BIOS 的功能性和能力的完全再实现。BIOS 通常在 Intel X86 计算机上以 16 位实模式操作，架构能够运行在 32 位或 64 位存储模式中，且不会被限制到特定的平台体系结构。同样，BIOS 通常几乎完全以汇编码编写，架构几乎完全以 C 编程语言编写。

架构提供 EFI(“可扩展固件接口”)的实现，EFI 是完全从固件的细节和实现提取的接口规范，该固件从装载和运行操作系统的软件来启动系统。EFI

被定义为完全不“透明”的接口，以使得可以在现有 BIOS 之上以及从暂存区创建 EFI 的实现、如架构创建。“不透明”在这里表示接口的访问者或客户对基础实现的内部不可见。访问者唯一的访问是通过接口本身。

EFI 引入了一组标准机制，即预启动组件用于相互交换，这也由架构使用，并且与本发明的不同实施例有关。部件之间的交互定义为“协议”，“协议”均为不透明的 C 语言 API。协议由“驱动程序”执行，驱动程序是能够分别载入存储器并被执行的单独组件。每个协议都具有 C 语言源代码“名称”和其自身唯一的标识符(这被实现为“GUID”或全球唯一标识符)，该标识符用于在执行时间将它与其它协议区分开。当驱动程序装载到存储器中时，它向 EFI 环境描述它所支持的一组协议。EFI 环境留意在存储数据库中的这些协议，让其它预启动代码搜索这些协议并访问它们，并在已经“输出”这些协议的驱动程序中找到代码并执行代码，无需知道特定驱动程序中协议的基础实现。

组件之间交互的不透明风格应当为那些精通 COM 或 CORBA 的人熟悉。特别是，在 COM 中也使用唯一标识符(GUID)来“查找”组件的接口。如同 COM 和 CORBA 一样，EFI 被设计成“可扩展的”。可以定义新的协议，并且可以创建新的现有协议的实现。在任一情况下，这些扩展以均匀方式适应基础环境。

驱动程序的概念对于那些精通当代操作系统的人是熟知的，其中许多不同类型的硬件需要以统一的方式由软件支持。在这些操作系统驱动环境中，对于不同类别的硬件装置定义了通用 API。这些通用 API 隐藏了基础硬件差异的细节，而实现相同 API 的不同驱动程序负责处理硬件变化。

除了由 EFI 定义的预启动组件之间的交互的标准机制，本发明说明性实施例支持仅在架构中定义的某些扩展。架构引入“固件卷(Firmware Volume)”的概念，这是为直接在 ROM 部件中的存储特别定义的平面文件系统。固件卷定义了与该文件系统交互的一组协议。在架构中，所有存储在 ROM 映像中的文件存储在固件卷中。由此在建立架构的过程期间产生固件卷。

在架构固件卷说明书中定义的扩展协议其中之一是“GUIDed Section Extraction Protocol”。该协议以这样的方式被定义，即厂商能够利用新定义的“段(section)”(这是文件的部分)扩展固件卷，这些段的表示是完全不透明的。该扩展发生的方式是固件卷中的文件的段头部包含 GUID，由固件卷代码使用 GUID 来定位实现“GUIDed Section Extraction Protocol”的驱动程序。通过协议调用

该驱动程序，并且随着文件从固件卷中被提取并放置在存储器中，传送不透明表示(opaqe representation)。可能传送的例子包括不透明表示的解密或对不透明表示检查数字签名。

虽然这里包含的许多例子提到架构环境，但应当理解本发明也可以利用更多传统的 BIOS 组件通过对架构描述进行调整来实现，该调整对于本领域的技术人员来说是显而易见的。

本发明的示例实施例在建立过程期间使用嵌入到 ROM 映像中的唯一标识符和消息摘要的组合，以安全识别未存储在 ROM 中的固件模块。其它传统解决方法提供主要对隐藏或安全的磁盘分区增加在预启动中可访问的固件的安全方法，但这些其它解决方法不要求在建立 ROM 映像的时候存在固件。SHA-1 或其它类似消息摘要算法的使用使得很难创建产生相同消息摘要的不同模块。由于消息摘要嵌入在 ROM 映像中，改变消息摘要本身需要更新 ROM 部件。一旦消息摘要已经嵌入到 ROM 映像中，存在多种现有方法检测或防止 ROM 映像的篡改，例如使闪存部件区域只读的 TPM 硬件和标记，这在本发明的范围之外但是对本领域的技术人员来说是公知的。因为消息摘要嵌入在 ROM 映像中，如果 ROM 映像是安全的并且不能被篡改，那么虚拟 ROM 模块实际上也是安全的并且不能被秘密地改变。递归方法也是可能的，因为虚拟 ROM 模块本身能包含指向存储在别处的其它虚拟 ROM 模块的“虚拟 ROM 指示器”。本发明的安全性对于所有虚拟 ROM 模块是可传递的，因此对于可用于安全预启动环境的固件数量几乎没有限制，只要所有的固件能在建立过程中可用即可。

为每个虚拟 ROM 模块使用唯一标识符也为虚拟 ROM 模块提供了单独的位置。每个虚拟 ROM 模块可以存储在多个位置中－在磁盘上、在 CD-ROM 上、在局域网上或在因特网上。因为消息摘要算法提供了验证模块的安全方法，所以位置本身或者将模块从存储位置其中之一传输到 PC 上的存储器的机制都不要求是安全的。

为计算机预启动提供附加固件的第一步骤是设计决定什么应该包括在实际 ROM 映像中并且什么应该从实际 ROM 映像中省去。从实际 ROM 映像中省去的信息被打包成一个或多个分开可访问的虚拟 ROM 模块。处于性能原因，沿其最通用路径启动 PC 所需的软件经常留在实际 ROM 中。然而，不常使用的

其它软件例如设置 UI、对其它语言的支持、不常用的启动选项、恢复或重新刷新固件以及许多可能的预启动应用和实用程序可以设置在实际 ROM 外部的虚拟 ROM 模块中。

虚拟 ROM 模块可以是架构“固件卷”，固件卷是组成包括许多文件的平面文件系统的单独的软件包。这些包括的文件的每一个可以是架构中支持的任一类型的文件。架构固件卷的优点是存在支持访问嵌入文件的架构固件(驱动程序)。然而，虚拟 ROM 模块可以是任何其它类型的文件，包括数据和可执行代码。如上所述，设计决定可以产生虚拟 ROM 模块的多级嵌套。换句话说，一些虚拟 ROM 模块可以包含指向其它虚拟 ROM 模块的虚拟 ROM 指示器。

对于每个这样的虚拟 ROM 模块，产生包括 SHA-1 消息摘要和 128 位 GUID(全球唯一标识符) 的虚拟 ROM 指示器。该消息摘要和 GUID 可以使用任何已知算法产生。产生的 GUID 广泛地用于架构中，但也同样用于计算的其它方面。GUID 和 SHA-1 消息摘要产生了具有极低复制概率的值。图 1 描述了 VROM 模块和产生的消息摘要和 GUID 之间的相应关系。对于第一 VROM 模块 2，包括 GUID 4 和消息摘要 6 的第一 VROM 逻辑指示器 20 与第一 VROM 模块相关联，并在第一 VROM 模块的基础上产生。对于第二 VROM 模块 12，产生包括 GUID 14 和消息摘要 16 的分开的唯一 VROM 逻辑指示器 22。

在为虚拟 ROM 模块产生 GUID 和消息摘要之后，根据标准 ROM 代码和虚拟 ROM 指示器建立实际 ROM 映像。在图 2 中示出该建立过程。指向第一 VROM 模块 2 的第一 VROM 逻辑指示器 20 包括产生的与第一 VROM 模块相关的 GUID 4 和消息摘要 6。指向第二 VROM 模块 12 的第二 VROM 逻辑指示器包括产生的与第二 VROM 模块相关的 GUID14 和消息摘要 16。第一 VROM 逻辑指示器 20 和第二 VROM 逻辑指示器 22 以及规则 ROM 代码 24 一起用于建立 ROM 映像 30。加入到 ROM 映像 30 的其它 ROM 代码 24 是设计者觉得最适合位于实际直接访问的 ROM 部件上的 ROM 代码。

从 ROM 映像 30 存储和检索虚拟 ROM 指示器 20 和 22 的规范依赖于具体实施例。在基于架构的一个实施例中，另一固件卷(其在 ROM 映像中)包含各种类型的“文件”。这些文件的每一个构建有多个段，其中每个段涉及文件的一些特殊属性。由于在固件卷中每个这样的文件都是由 GUID 唯一标识的，这个 GUID 可以是与用于检索虚拟 ROM 模块的标识符相同的标识符，并且该

虚拟ROM模块变成架构卷中的一个“文件”。可使用“Guided Section Extraction Protocol”实现另一这样的段。该段包含识别进行提取所需的驱动程序的GUID和验证内容的SHA-1消息摘要(已经包括作为文件的标识符的GUID)。另外，其它表示也可以在架构中或作为现有BIOS的一部分被构建，只要提供GUID和SHA-1消息摘要。

图3示出如何将实际ROM映像放入ROM中，并示出如何在别处分布虚拟ROM模块。将包括第一VROM指示器20、第二VROM指示器22和其它ROM代码24的ROM映像30放入ROM部件40、如闪存。第一VROM指示器指向的实际第一VROM模块2可位于与ROM部件40相同的电子装置上的硬磁盘42中。第二VROM指示器指向的第二VROM模块12可以存储在可通过因特网44或其它类型网络访问的位置46上。本领域技术人员可以理解，将VROM模块分布到其它存储位置也是可能的，这也在本发明的范围之内。应当注意，分布事件不一定都同时发生。虽然实际ROM映像对于启动PC是必需的，但虚拟ROM模块仅以一些其它方式在某些其它时间点可用。也不需要用于虚拟ROM模块的单个分布机制。每个这样的虚拟ROM模块可采用各种不同方式分布。

分布机制的一个例子可适用于升级计算机中的硬盘。在固件建立期间，首先使将驱动内容从一个硬盘拷贝到另一硬盘的固件可用。然而固件没有构成实际ROM的一部分，但是指向它的实际ROM指示器存储在实际ROM映象中。在磁盘升级的时候，这可能是许多年以后，相应的虚拟ROM模块在必须用来执行更新的CDROM上是可用的。由于文件先前已经“包括”在建立过程中，因而它们能够被相信并从CD-ROM运行。可选择地，拷贝软件的硬盘驱动程序可以从因特网下载，并以同样的方式验证。在建立PC的磁盘映像期间，其它分布机制可以是合适的。磁盘映像可以包括虚拟ROM模块能被拷贝到的非安全磁盘分区。

图4是接着本发明示例实施例在建立ROM映像的步骤期间使用虚拟ROM模块的步骤顺序的流程图。建立在步骤50开始，确定是否存在更多虚拟ROM模块(步骤51)。如果存在虚拟ROM模块，则产生GUID并指定给虚拟ROM模块(步骤52)。然后产生用于虚拟ROM模块的消息摘要、如SHA-1消息摘要(步骤54)。然后根据GUID-消息摘要对构造虚拟ROM逻辑指示器(步骤56)。

重复该步骤直到确定没有更多的虚拟 ROM 模块(步骤 51)。然后组合包括虚拟 ROM 逻辑指示器和规则 ROM 代码的 ROM 映像(步骤 58)，并且建立过程结束(步骤 60)。

一旦分布了虚拟 ROM 模块，它们可以被检索到 PC 的存储器或 ROM 部件 40 所位于的其它电子装置中。图 5 示出检索过程的视图。ROM 映像 30 装载到 PC 存储器 70 中。在 ROM 映像 30 中的 VROM 逻辑指示器 20 和 22 指出存储在硬盘 42 和存储位置 46 中的第一和第二 VROM 模块 2 和 12 的存在和身份。如下面进一步描述的，该第一和第二 VROM 模块 2 和 12 然后被检索并被装载到存储器 70 中。可能存在检索特定 VROM 模块的各种原因，包括执行知道需要装载特定虚拟 ROM 模块的一些其它固件，或与一些接口交互的使用者想要一些不在实际 ROM 中的固件被执行。

在基于架构的实施例中，其中虚拟 ROM 指示器作为段存储在固件卷中，检索 VROM 模块的决定实现为将“文件”“装载到存储器”的决定，该“文件”指段。在这个实施例中，支持“Guided Section Extraction Protocol”的如前所述的架构驱动程序实现图 6 中的流程图的方法。应当理解，虽然该描述指的是试图检索虚拟 ROM 模块作为“检索驱动程序”的固件，但该固件不可以作为非架构实施例中的驱动程序来实现。

检索驱动程序能够使用存储在非易失性存储器中的变量，该变量描述了“到哪里找”驱动程序的搜索路径。通常在操作系统中使用搜索路径以提供能在存储器中找到、装载和执行的称为可执行程序的方法。虽然存在能够支持这种搜索的多种实现机制，但根据本发明的搜索路径的使用是类似的。在架构实施例中，变量作为可被用户或一些其它软件在任何时候更新的 EFI NVRAM 变量存储在 ROM 部件。

检索驱动程序依次迭代通过搜索路径元素。该搜索路径表示搜索虚拟 ROM 模块的“优选顺序”。对于在搜索路径中的每一个这种元素，该驱动程序试图检索使用那个元素的虚拟 ROM 模块。每个元素可以编码为“URI 模板”，其中，作为在浏览器和因特网中广泛使用的通用资源标识符(或通用资源定位器)不完全地指定相应文件的位置。该不完全的指定可以仅是利用与虚拟 ROM 模块相关的 GUID 填充的变量。一旦填充了，这种“完全的”URI 可以指在本地磁盘分区上的一些目录中的文件、能够使用超文本传输协议(HTTP)或一些其它

公知协议在因特网上检索到的文件。同样，在架构中，URI 中的每个“方案”的检索机制(识别“file:”“Http:”“ftp:”等的部分)本身可以交给理解特定协议的不同架构驱动程序。

具有任何特定元素的检索可能不会找到相应的虚拟 ROM 模块，在这种情况下企图检索搜索路径中的下一元素。如果检索企图成功，则虚拟 ROM 模块被装载到存储器中。在这种情况下，通过运行与构建过程期间使用的算法相同的消息摘要算法对虚拟 ROM 模块的消息摘要进行确认。如果虚拟 ROM 模块的消息摘要匹配，则虚拟 ROM 模块是可用的。然而，如果消息摘要不匹配，则检索进行到搜索路径的下一元素。

在确认消息摘要后，作出虚拟ROM模块是否可以“更接近于”实际ROM的决定。“更近”的定义可以是在留出用于高速缓存虚拟 ROM 模块的本地磁盘驱动器上的标准位置。如果存在这样的位置，并且该位置在搜索路径中比虚拟 ROM 模块刚被检索的位置指定的更早，则在该虚拟 ROM 模块被访问前将其从存储器写到这个位置。

图 6 是接着本发明示例实施例检索虚拟 ROM 模块的步骤顺序的流程图。检索步骤在 80 开始，确定在搜索路径中是否存在更多元素(步骤 81)。如果在搜索路径中存在附加元素(步骤 81)，则从搜索路径中选出下一位置(步骤 82)。然后使用虚拟 ROM 模块的 GUID 试图将虚拟 ROM 模块检索到存储器中(步骤 84)。如果检索成功，则产生用于虚拟 ROM 模块的 SHA-1 消息摘要(步骤 86)并将其与存储的虚拟 ROM 模块比较(步骤 88)。如果消息摘要匹配(步骤 89)，则作出虚拟 ROM 模块是否可以被高速缓存地更近的决定(如上所述)(步骤 91)。如果虚拟 ROM 模块可以被高速缓存地更近(步骤 91),则它在搜索路径中就会被高速缓存地更近(步骤 92)。然后将虚拟 ROM 模块装载到它可以被访问的存储器中并且检索步骤结束(步骤 96)。

本发明的示例实施例也可用于提供用于安全地对固件提供更新的机制。修改上述产生唯一标识符和消息摘要的步骤包括产生两个新的属性一版本号和用于更新验证过程的唯一标识符。例如，在架构实施例中，架构卷“文件”可以包含版本号的段一具有可选提供文本串的单调增长数值。同样，架构“Guided Section Extraction Protocol”包含涉及支持段提取的驱动程序示例的 GUID。换句话说，可使用对下载和确认虚拟 ROM 模块的内容负责的分离驱动程序。该驱

动程序可以调用一组嵌套的驱动程序。

图 7 示出需要为每个虚拟 ROM 模块封装的附加属性。VROM 模块用于产生 VROM 指示器 110。VROM 指示器 110 包括 GUID 112 和 SHA-1 消息摘要 114。另外，VROM 指示器 110 也包括版本号 116 和唯一识别更新验证过程的更新验证标识符 118。图 8 示出用于模块的相应虚拟 ROM 指示器 110 是怎样与“更新验证器”代码 120 以及实际 ROM130 的其它部分一起封装到实际 ROM 映像 140 中的。应当注意，并不要求“更新验证器”模块 120 存储在实际 ROM 140 中一它可以作为附加虚拟 ROM 模块存储。同样，可以在架构中实现作为在 Guided section Extraction 协议驱动程序上的变量的更新验证代码(然而，在架构实现中，至少一组 Guided Section Extraction 协议驱动程序必须存储在实际 ROM 中。

在一个实施例中，使用 PGP 类解决方法用于确认。(PGP 或“优良保密协议”是一组用于数字签名和/或对电子邮件消息进行加密的算法，这对于本领域技术人员来说是公知)。这个过程在图 9 中示出。在固件建立时，对于特定的“更新验证器”模块，通过密钥产生过程 150 产生公私数字密钥对。用于该过程的算法可以是任何标准算法不对称密钥算法、例如 RSA。利用更新验证器模块 160 存储公钥 152，固件厂商将私钥 154 存储在其自己的单独位置 170 中。下面讨论用于更新验证的附加备选安全机理。

图 10 的流程图示出在为一个或多个虚拟 ROM 模块产生更新时私钥的使用。为了利用本发明示例实施例产生更新，在更新准备好的时候 200，厂商使用其私钥以产生更新“模块密钥”。厂商代码对新模块(新模块可包括版本号)运行例如 SHA-1 的算法(步骤 202)，然后使用公-私密钥的私钥将所得的消息摘要加密(步骤 204)。成功产生导致由私钥标记的加密的更新模块密钥 206。本领域技术人员可以知道类似的加密机制可以在其它实现中使用。

一旦产生更新，则更新可以被检索、验证然后被“修补”到虚拟 ROM 模块中。检索步骤需要版本检查。特定虚拟 ROM 模块的先前版本可以已经被下载并在“本地高速缓存”中可用。使用许多公知机制中的一种来确定何时以及如何检查较新的版本。该机制留给更新验证器来实现。例如，更新验证器可以周期地进行较新版本的验证。可选地，特定位置可“预留”用于更新，并总是被检查。这些更新可以由操作系统运行的代码来提供。在没有较新版本存在的

情况下，进行上述在图 6 中描述的过程。不考虑检查更新的机制，如果检测到新版本，则它必须被验证。这个验证过程从根本上不同于上述用于原始虚拟 ROM 模块的验证过程，因为更新的虚拟 ROM 模块在建立虚拟 ROM 的初期是不可用的。

在本发明的一个实现中，与更新相关的“模块密钥”如图 11 和 12 中所述的那样被验证。使用公-私密钥的公钥将“模块密钥”解密，并利用更新验证器存储。然后模块本身被散列，比较两个结果是否相等。

图 11 提供了该步骤的一个可能实现的综述，其中已经装载到 PC 存储器 200 中的 ROM 映像 210 中的虚拟 ROM 指示器 220 逻辑地指向更新的虚拟 ROM 模块 250。该更新的 VROM 模块 250 可以存储在可通过因特网 255 访问的厂商位置 260 中。更新的 VROM 模块 260 被下载到 PC 存储器 200 中并由更新验证器 230 使用公钥 235 验证，然后存储在本地硬盘 270 上。如前所述，虚拟 ROM 指示器的“修补”需要虚拟 ROM 指示器本身能够在这个过程期间被更新。出于安全的原因，更新过程也需要未授权方在处理中不能更新虚拟 ROM 指示器。而且，如上所述，如果原始虚拟 ROM 建立过程是安全的，并且存在用于防止后来未授权方更新 ROM 的一些机制，那么这个更新过程是非常安全的，因为为了进行更新仅执行授权的代码。一旦验证过程成功，更新验证器 230 更新存储在实际 ROM 映像 210 中的虚拟 ROM 指示器 220，以使得它与新模块版本 250 一致，然后可选地在本地磁盘 270 上高速缓存更新的模块。更新涉及改变利用虚拟 ROM 指示器 220 存储的版本号和 SHA-1 消息摘要。

图 12 的流程图进一步示出更新过程。更新检索过程从 280 开始，确定是否存在未搜索的更新位置(步骤 281)。如果存在未搜索的更新位置(步骤 281),则选择搜索路径中的下一位置(步骤 282)并使用更新模块 GUID 试图把较新的模块检索到存储器中(步骤 284)。如果新版本存在(步骤 285)，则使用参考图 10 所述的模块密钥验证过程来验证模块(步骤 288)。如果不存在 VROM 模块的新版本，则随后是标准检索处理(步骤 286)。在存在较新的模块的情况下，试图验证(步骤 289)。在验证成功的情况下，如上所述确定新模块是否可以被高速缓存地“更近”(步骤 291)。如果新模块能够被高速缓存地更近，则它在搜索路径中被高速缓存地更近(步骤 292)。不管高速管缓存，验证器过程更新 VROM 指示器以反映更新的模块(步骤 294)，然后访问存储器中的更新的模块(步骤

296)，在该点检索过程结束298。

图 13 中概括了接着本发明示例实施例用于验证与更新模块相关的模块密钥的步骤顺序。模块密钥验证过程从 300 开始，产生关于新模块的 SHA-1 消息摘要(步骤 302)。本领域技术人员知道，在不脱离本发明范围的情况下，不同的实现可以使用用于这里讨论的加密和解密过程的不同消息摘要算法。然后使用公-私密钥的公钥将该模块密钥解密(步骤 304)。比较这两个消息摘要以确认它们是否相等(步骤 305)。如果比较显示是相等的消息摘要，那么验证 306 为成功的306。如果比较没有显示相同的消息摘要，那么验证就是失败的308。

如上面提到的，备选确认技术也可用作更新过程的一部分。只要未授权的读取不能访问 ROM 的一部分，则对称私钥可用于更新验证。下面将讨论这样作的技术。对于该技术，在建立过程中产生公共密钥，并由 ROM 以及厂商以加密形式存储。可以使用相同的对称密钥对更新“密钥”进行加密和解密。可选择地，可以利用可信任的服务器创建使用 TLS 或 SSL 的通信的安全隧道来下载更新。该验证以通信发生在可信任方为基础，而不是以利用更新本身传输的消息摘要为基础。

本发明的示例实施例也可用于创建和访问用于固件的安全存储。通过保持隐藏在 ROM 映像内的秘密密钥用于加密或签名，建立没有位于 ROM 映像中的安全存储块。可采用与上述任何虚拟 ROM 模块被访问和引用的方式相同的方式访问和引用安全存储，但如果存储区是在除本地磁盘之外的其它地方时，需要附加的唯一机器标识符。

在架构环境中，单个固件卷可用于保护、加密的存储，所述存储用于希望使用该存储的所有固件代码。固件卷的使用意味着从非 ROM 位置读取作为单个单位的存储，并作为单个单位写回非 ROM 位置。使用多个固件卷的其它解决方法也立即变得清楚了。这些附加固件卷可以要么由实际 ROM 访问，要么由第一固件卷以如上所述的递归方式访问。同样，每个这样的固件卷可以具有识别它是否被加密或不能写入的标记。

本发明的示例实施例可以在建立过程期间预留存储。在虚拟 ROM 的建立过程期间，特定文件被标记成需要加密的存储。这些文件的每一个担当由固件代码的特定块最终使用的一些加密存储区的占位符。建立过程将所有这些占位符结合到固件卷中，使用 SHA-1 算法产生用于固件卷的消息摘要，并在用于

ROM 部件的主固件卷中存储对这个固件卷的参考(或虚拟 ROM 指示器)。如前所述，这个虚拟 ROM 指示器包含用于参考的(加密存储)固件卷的唯一标识符和指示访问驱动器以检索固件卷的 GUID。另外，它包含两个标记，一个示出它应当在运行时间被加密，第二个示出它还没有被加密。

用于实际 ROM 的建立过程也预留用于两个附加变量的空间：机器标识符和加密密钥。用于机器标识符的空间设置在标记为在预启动期间不可写入的固件卷中，而用于加密密钥的空间放置在标记为在预启动期间不可读取的固件卷中(将在下面详细描述该不可读取和不可写入)。随后，预启动代码可以决定检索第一次包含加密存储的固件卷。在考虑虚拟 ROM 指示器时，预启动代码确定访问的文件被标记为要加密，但是还没有被加密。在这种情况下，预启动代码使用上述相同机制检索文件的第一版本。由于还没有使用存储，因而在建立时间构造的相应固件卷的通用例子对于预启动代码的所有运行例子都是可用的。如上所述，将固件卷像前面一样装载到存储器中并以与其它固件卷相同的方式使固件卷对于预启动代码可访问。然而，提供访问该固件卷的驱动程序也留意“脏(dirty)”标记(指出它应当写回到持久存储媒介)以及虚拟 ROM 指示器的位置，因为后者需要被更新。在该固件卷中的文件的任何更新会导致设置脏标记。

图 14 示出一旦安全存储区(即实施安全存储区的固件卷)第一次被读取并且已经产生了机器 ID 和加密密钥，存储器 320 是如何分解的。存储区 320 包括 ROM 映像 330 和安全存储区 340。该 ROM 映像包括可更新区域 332 和防止被更新的(写保护的)区域 334。防止被更新的区域 334 保存安全存储 VROM 指示器 335 和用于唯一机器 ID 336 的存储区。该安全存储 VROM 指示器包括用于 VROM 模块的 GUID、消息摘要 337 和加密标志 333。该 ROM 映像也具有包括用于加密密钥 339 的存储区的防止被读取的(读保护的)区域 338。

在一些时间点(虽然对固件卷的每次写入并不是必要的)，预启动代码决定把修改后的固件卷写入到一些存储媒介。写入决定由前面段中描述的脏标记来驱动。预启动代码检测到还没有产生加密密钥和机器标识符。它产生机器标识符和加密密钥并将它们保存在实际 ROM 中它们各自的存储位置。可以使用任何用于产生唯一值的适当机制以产生这些标识符。一个解决方法是对两者都使用 GUID。

在产生标识符之后，本发明的示例实施例然后在虚拟 ROM 指示器中检测应当被加密但还没有被加密的固件卷。它使用加密密钥对固件卷加密并对结果产生 SHA-1 消息摘要。对本发明的目的来说，可以使用任何适当的加密算法，包括 Blowfi sh、DES/3DES、IDEA 等。

在对固件卷加密并对结果产生消息摘要后，预启动代码然后必须确定在哪写入加密的固件卷。用于加密的固件卷的一个位置是在本地磁盘驱动器 350 上的一个目录，如图 15 中所示。然而，加密的固件卷 340 可以选择性地被存储在本地网络服务器上、在万维网或在其它可访问的位置。使用机器识别码 336 和识别固件卷文件的 GUID 331 的组合以把文件写入唯一位置。此时，预启动代码更新在实际 ROM 中的虚拟 ROM 指示器 335，表明固件卷 340 已经被加密了，并反映新产生的消息摘要 337。

图 16 的流程图中示出写入安全存储的过程。写入安全存储的过程从 400 开始，确定是否设置了脏标记(步骤 401)，该标记表示固件卷应当被写回到持久媒介。如果没有设定脏标记，则该过程结束 420。如果设置了脏标记，确定 VROM 指示器 335 中是否设置了加密标记 333(步骤 403)。如果设置了加密标记(步骤 403)，则使用加密密钥对安全存储区域进行加密(步骤 408)。如果没有设置加密标记(步骤 403)，则产生并存储机器 ID(步骤 404)，以及产生并存储加密密钥(步骤 406)。然后使用加密密钥对安全存储加密(步骤 408)。

在对安全存储加密(步骤 408)之后，对加密的安全存储产生消息摘要(步骤 410)。然后使用机器 ID 336 和存储 GUID 331 将该加密的存储写入到第二存储(步骤 412)。在写入安全存储(步骤 412)之后，利用新消息摘要和加密的标记更新 VROM 指示器(步骤 414)，并且写入过程结束 420。

图 17 的流程图示出本发明的示例实施例支持检索加密固件卷的预启动代码。读取安全存储过程从 440 开始，预启动代码确定 VROM 指示器中是否设置了加密标记(步骤 441)。如果预启动代码根据虚拟 ROM 指示器 335 中的标记检测出已经对固件 340 加密(步骤 441)，则它利用机器标识符 336 和 GUID 331 的组合检索和读取固件卷(步骤 444)。如果 VROM 指示器中没有设置加密标记，则预启动代码使用存储 GUID 从其它位置读取未加密的安全存储初始值(步骤 442)。一旦读取了固件卷(步骤 442 或步骤 444)，对安全存储产生消息摘要(步骤 446)。将产生的消息摘要与 VROM 指示器中的消息摘要进行比较和验证(步

骤 447)。如果不匹配，则安全存储读取失败(步骤 448)。如果验证成功，则预启动代码对固件卷解密(步骤 450)，并使得它在存储器 320 中可用，读取过程结束 460。如上所述，保持脏标记以确定是否对固件卷 340 作了修改。除了加密密钥和机器标识符不需要被重新创建之外，修改后的固件卷以与初始写入相同的方式被第二次或连续重复地写出存储。

在预启动处理中的某一点，本发明的示例实施例锁定了实际 ROM 的多个部分以防止进一步的篡改或访问。用于该目的的精确机制取决于硬件。该描述集中在该锁定发生时的特定时间点。该时间点可以是与修改后的固件卷通过被写入存储被冲刷(flush out)同时或之后。

不可读存储的一个 EFI 兼容时间点是在“退出启动服务”期间，这是 O.S.装载程序通知 EFI 环境将要进行装载操作系统的时间点。而这是非常易于实现的(可以在退出启动服务期间通知驱动程序)，这里的风险在于 O.S.装载程序代码的信任级。如果 O.S.装载程序代码作为固件建立过程的一部分以与其它虚拟 ROM 模块相同的方式一样是可用的，那么该解决方法是非常安全的。另一方面，如果以其它方式、例如使用逻辑启动协议访问 O.S.装载程序代码，则认为不可读实际 ROM 存储的最佳时间点只是在启动选项之前。在架构技术中这被称作在“DXE”(驱动运行环境)和“BDS”(启动装置选择)之间的转变。

本发明的示例实施例也允许预启动阶段产生数据，并以安全方式使数据对其它环境可用。两个文件在固件映像的建立过程期间保存在两个单独的安全存储固件卷中。第一文件专供公-私密钥的公钥之用，并保存在不可写的安全存储固件卷中。第二文件专供公-私密钥的私钥之用，并保存在不可读的安全存储固件卷中。

在预启动时，产生用于特定平台的唯一公-私密钥，并将其存储在各自的安全存储固件卷中。用于这个的算法可以是任何标准算法对称密钥算法例如 RSA。随后，通过对内容运行消息摘要算法，并使用公-私密钥的私钥对该内容加密，预启动固件标记创建的用于递送到其它环境的内容。内容和所得加密的消息摘要都可以存储在任何存储位置。当另一环境接收预启动内容时，公-私密钥的公钥存储在不可写的存储器中，但可以被读取。使用该公钥对内容附带的消息摘要解密，然后把这个结果与重新应用到内容的相同消息摘要算法的输出结果进行核对。这个过程确保内容没有被修改。由于其它环境不可访问公-私密钥的私钥，

私钥实际上是不可能以公-密钥的公钥用于解密和验证内容的方式产生或标记其它内容。

当然在不偏离本发明范围的情况下，可以进行一定的变化，在上面说明书中包含的或附图中示出的内容仅作为解释，并不拘泥于字面。本领域技术人员将会清楚，在附图中描述的步骤顺序和结构在不偏离本发明范围的情况下可以进行修改，并且这里包含的说明仅仅是本发明多个可能描述中的单个例子。

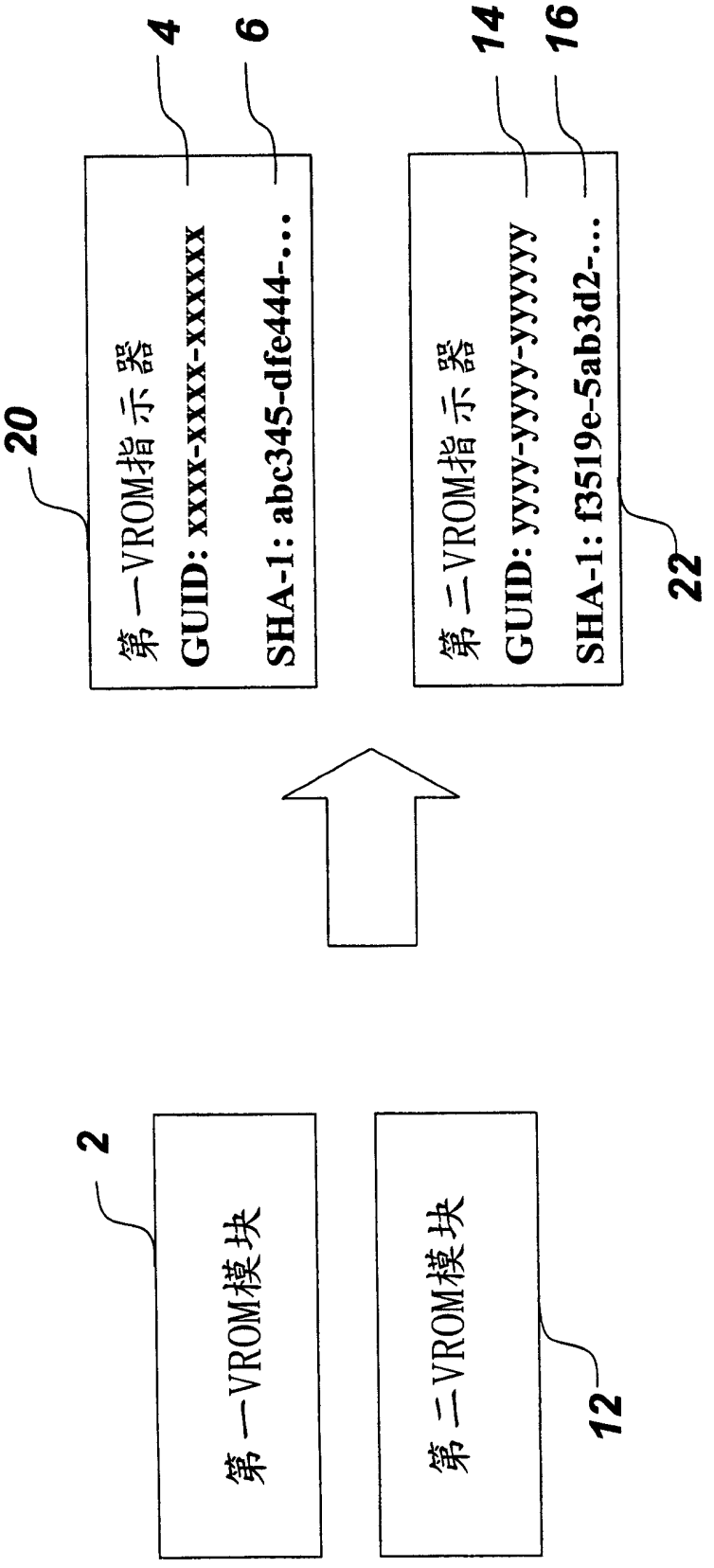

第一VROM模块
2
第二VROM模块
12
第一VROM指示器
GUID: xxxx-xxxx-xxxxxx
SHA-1: abc345-dfe444-...
4
6
20
第二VROM指示器
GUID: yyyy-yyyy-yyyyyy
SHA-1: f3519e-5ab3d2-...
14
16
22

图 1

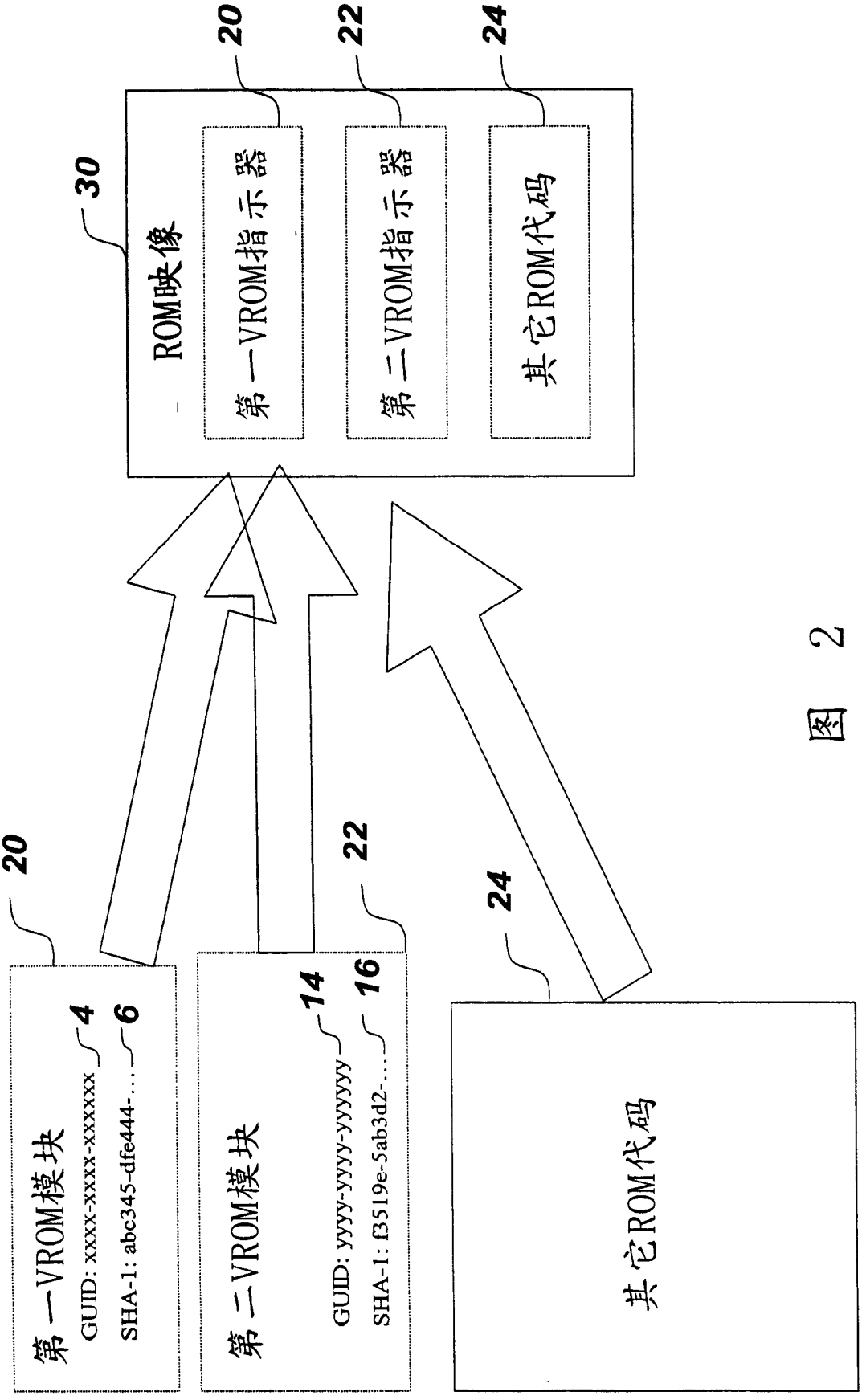

30
ROM映像
20
第一VROM指示器
22
第二VROM指示器
24
其它ROM代码
20
第一VROM模块
GUID: xxxx-xxxx-xxxxxx
4
SHA-1: abc345-dfe444-...
6
22
第二VROM模块
GUID: yyyy-yyyy-yyyyyy
14
SHA-1: f3519e-5ab3d2-...
16
24
其它ROM代码

图 2

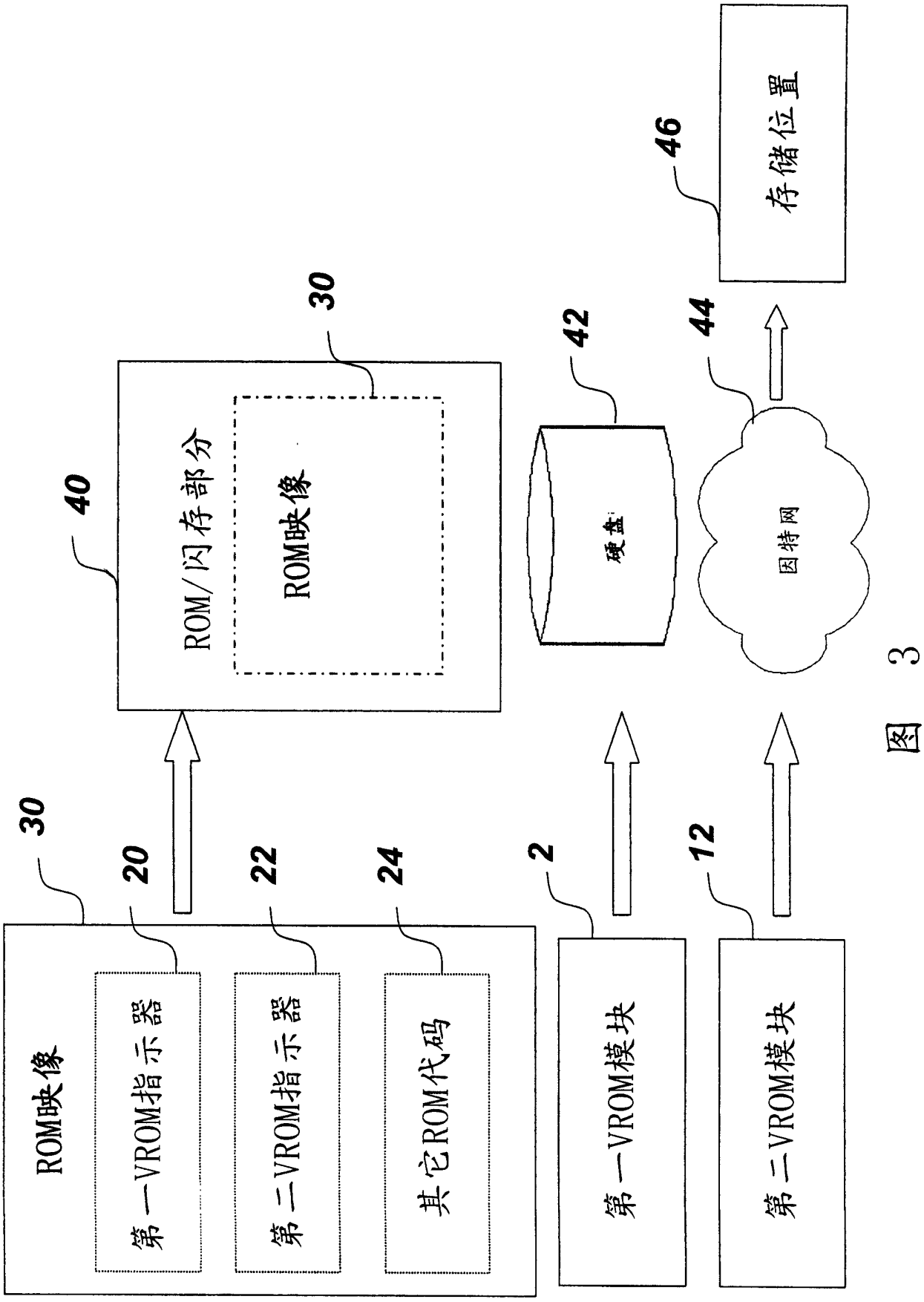

ROM映像
第一VROM指示器
第二VROM指示器
其它ROM代码
第一VROM模块
第二VROM模块
ROM/闪存部分
ROM映像
硬盘
因特网
存储位置
30
20
22
24
2
12
40
30
42
44
46

图 3

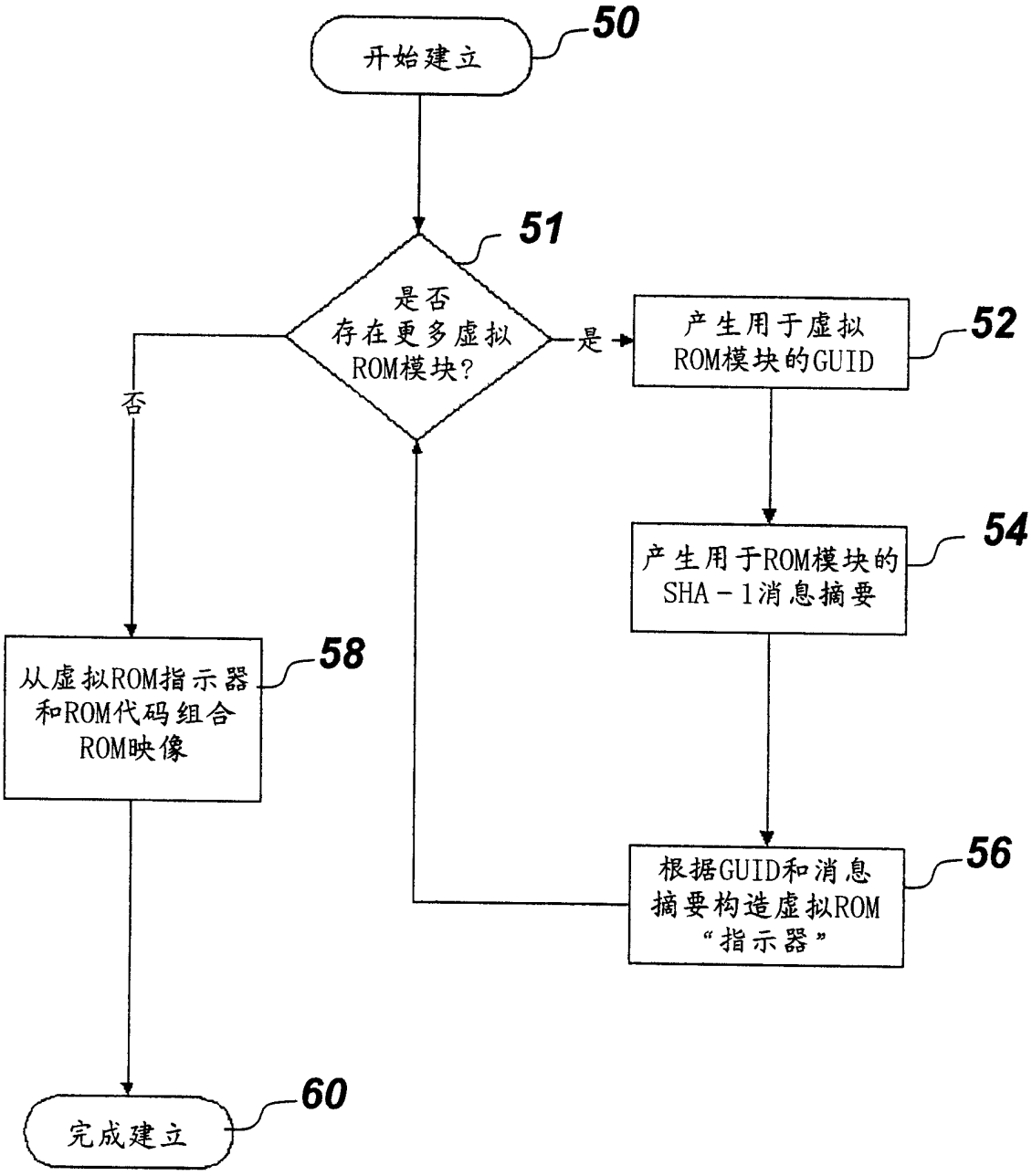

开始建立
50
是否
存在更多虚拟
ROM模块?
51
是
否
产生用于虚拟
ROM模块的GUID
52
产生用于ROM模块的
SHA－1消息摘要
54
根据GUID和消息
摘要构造虚拟ROM
“指示器”
56
从虚拟ROM指示器
和ROM代码组合
ROM映像
58
完成建立
60

图 4

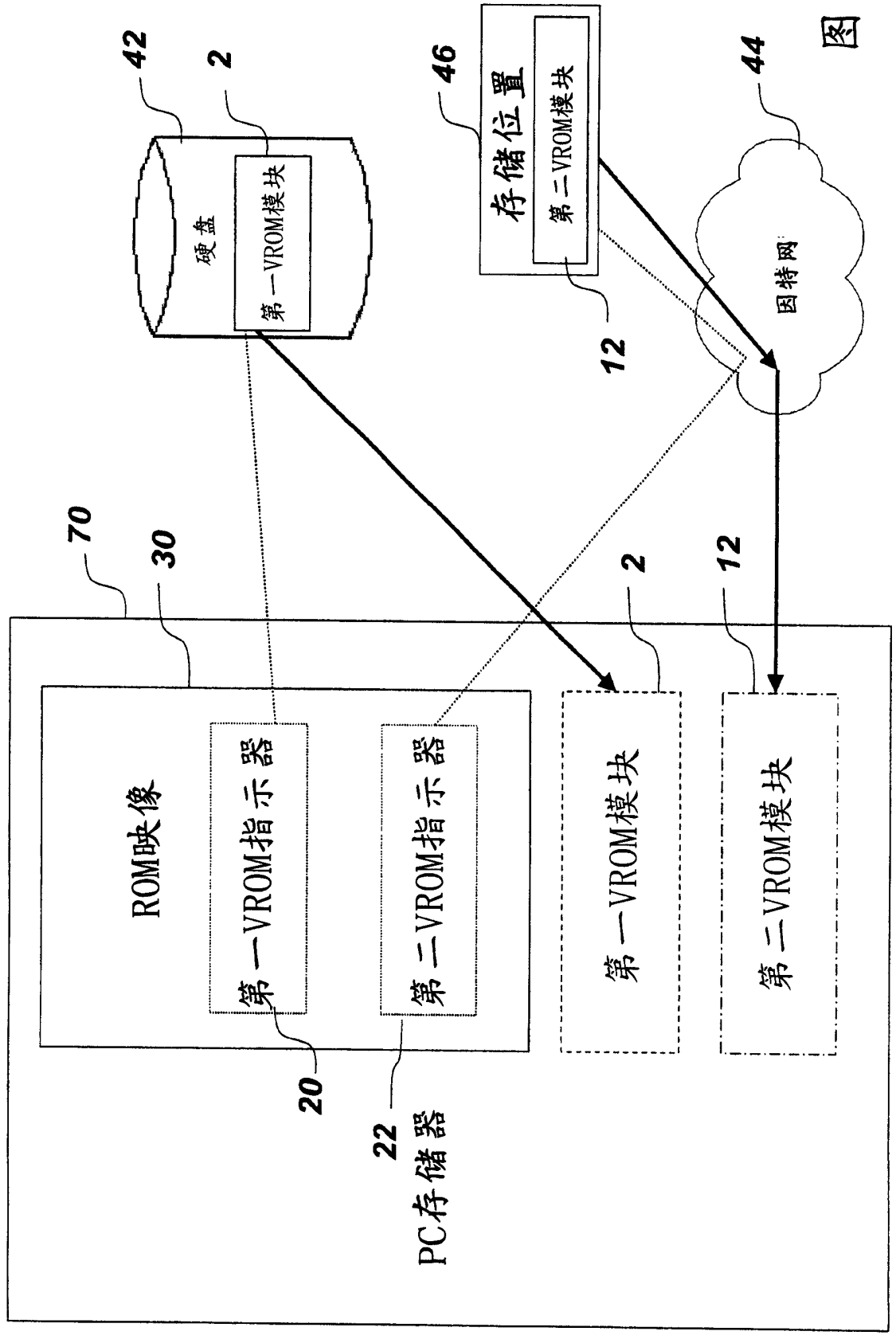

42
2
硬盘
第一VROM模块
46
存储位置
第二VROM模块
12
44
因特网
70
30
ROM映像
第一VROM指示器
第二VROM指示器
20
22
PC存储器
2
第一VROM模块
12
第二VROM模块

图 5

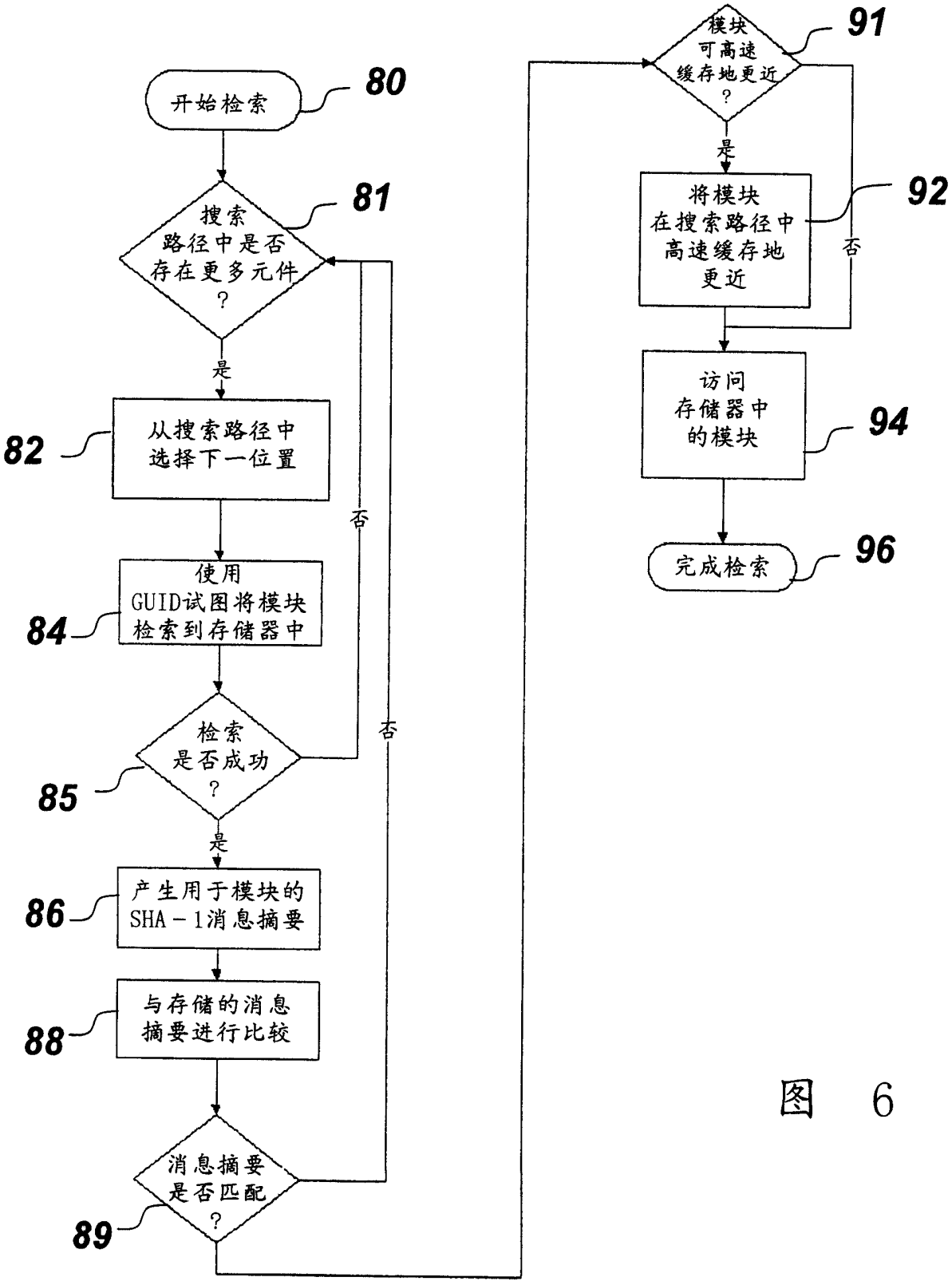

开始检索
80
搜索路径中是否存在更多元件？
81
是
否
从搜索路径中选择下一位置
82
使用GUID试图将模块检索到存储器中
84
检索是否成功？
85
是
否
产生用于模块的SHA－1消息摘要
86
与存储的消息摘要进行比较
88
消息摘要是否匹配？
89
模块可高速缓存地更近？
91
是
否
将模块在搜索路径中高速缓存地更近
92
访问存储器中的模块
94
完成检索
96

图 6

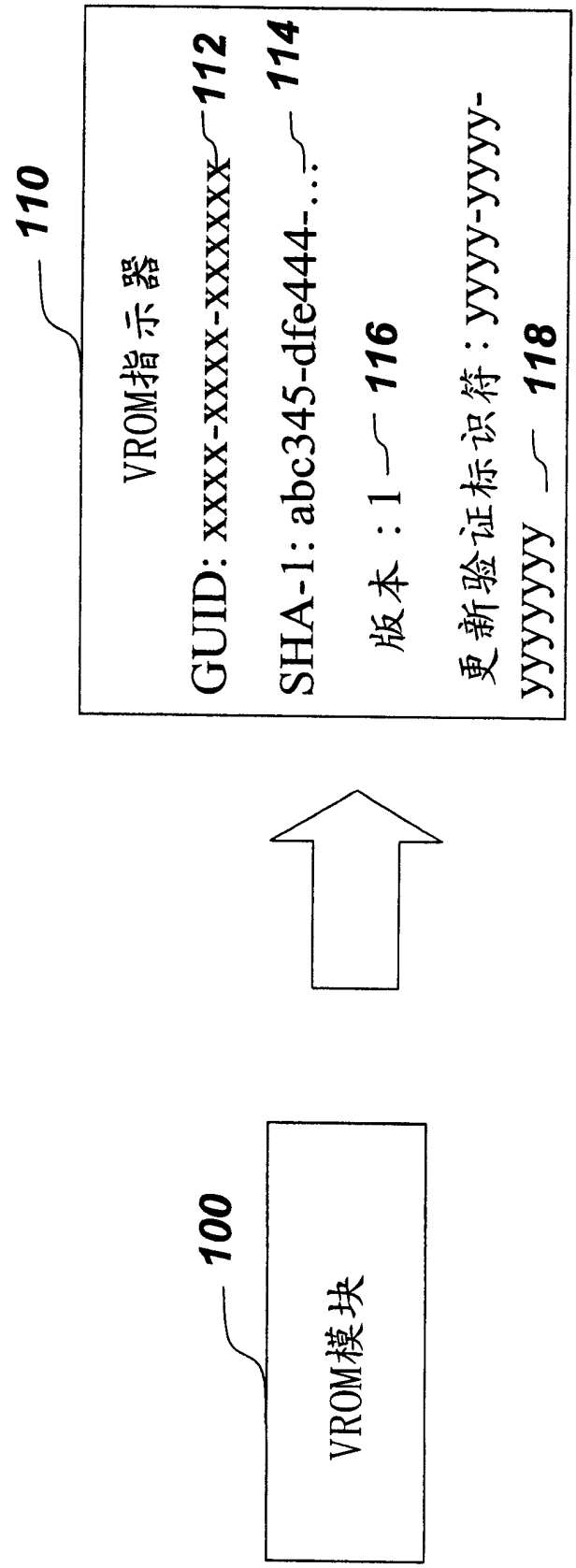

100
VROM模块
110
VROM指示器
GUID: XXXX-XXXX-XXXXXX
112
SHA-1: abc345-dfe444-...
114
版本：1
116
更新验证标识符：yyyy-yyyy-yyyyyyyy
118

图 7

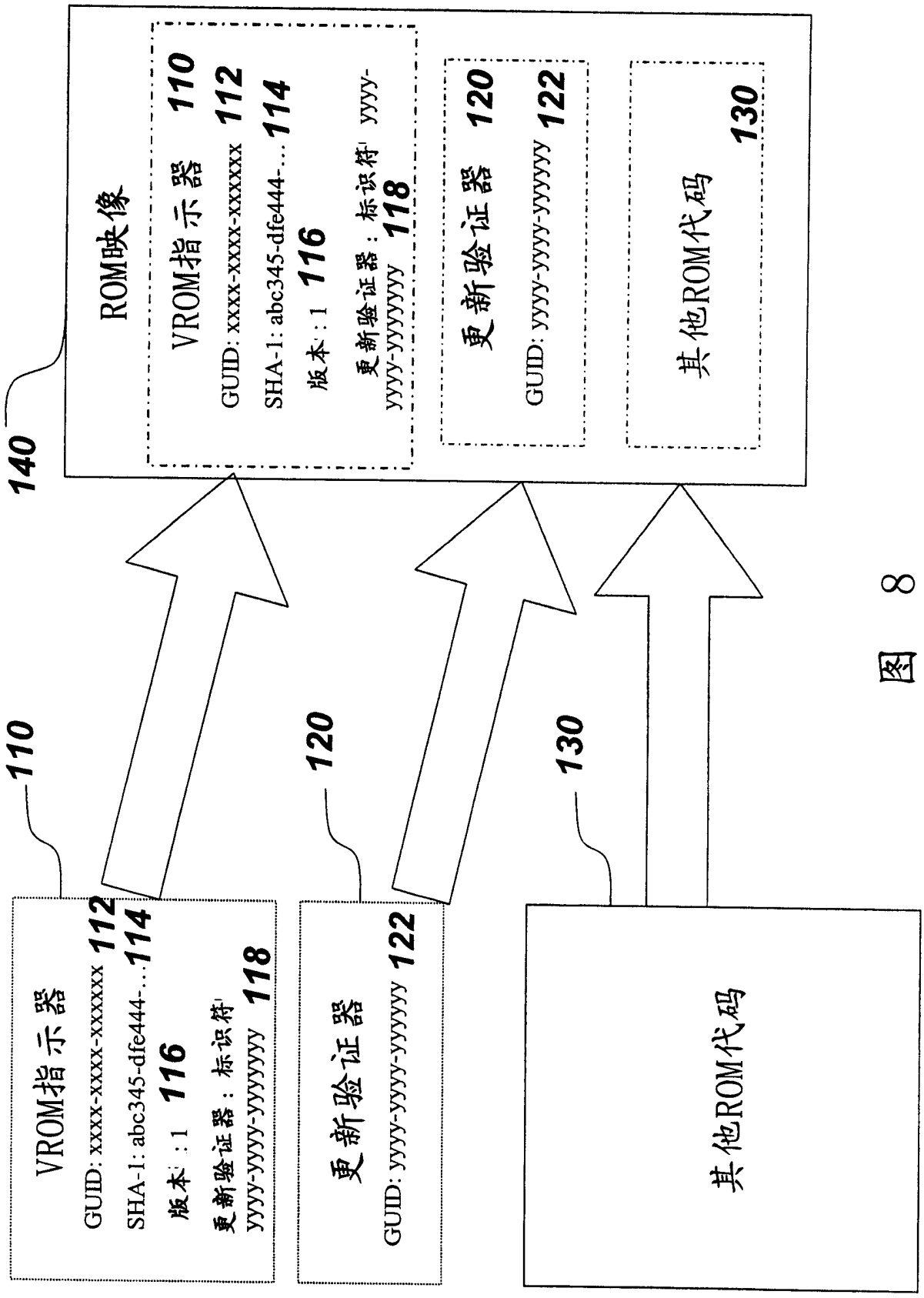

140
ROM映像
VROM指示器 110
GUID: xxxx-xxxx-xxxxxx 112
SHA-1: abc345-dfe444-... 114
版本: 1 116
更新验证器：标识符: yyyy-yyyy-yyyyyy 118
更新验证器 120
GUID: yyyy-yyyy-yyyyyy 122
其他ROM代码 130
110
VROM指示器
GUID: xxxx-xxxx-xxxxxx 112
SHA-1: abc345-dfe444-... 114
版本: 1 116
更新验证器：标识符: yyyy-yyyy-yyyyyy 118
120
更新验证器
GUID: yyyy-yyyy-yyyyyy 122
130
其他ROM代码

图 8

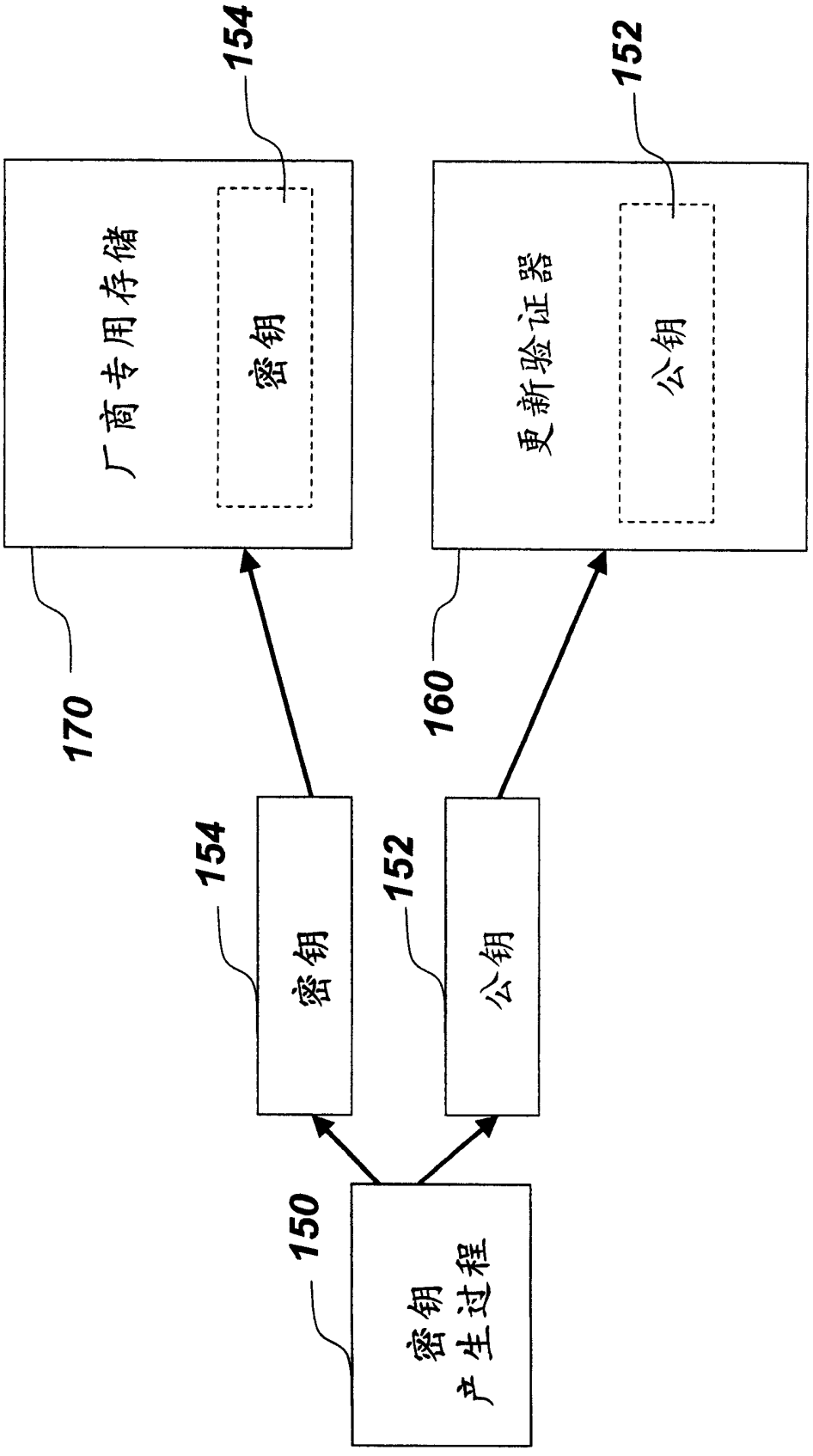


图 9

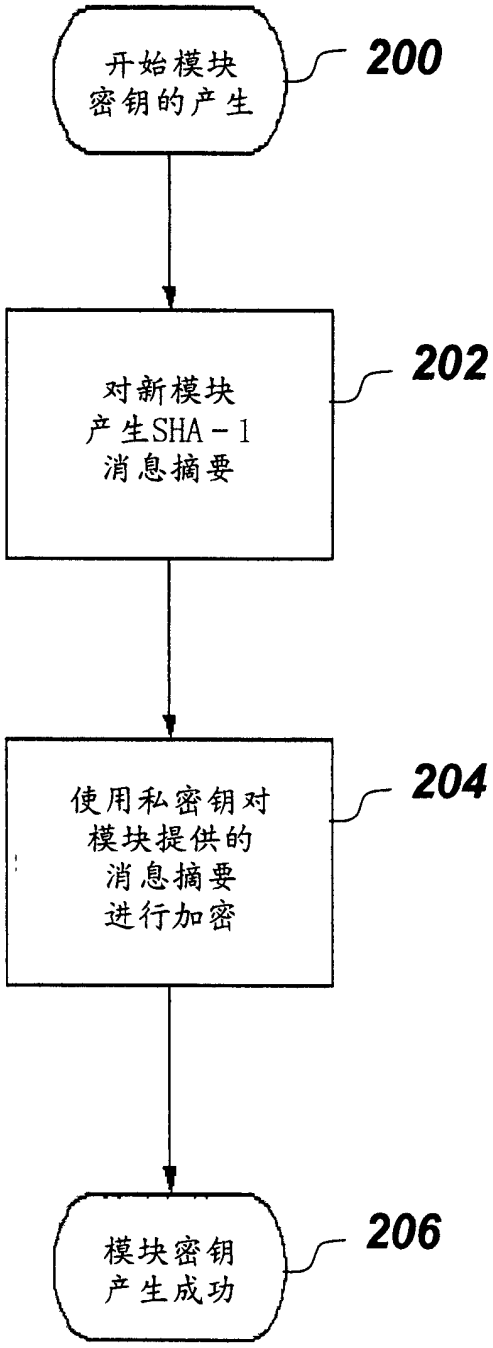

开始模块
密钥的产生
200
对新模块
产生SHA－1
消息摘要
202
使用私密钥对
模块提供的
消息摘要
进行加密
204
模块密钥
产生成功
206

图 10

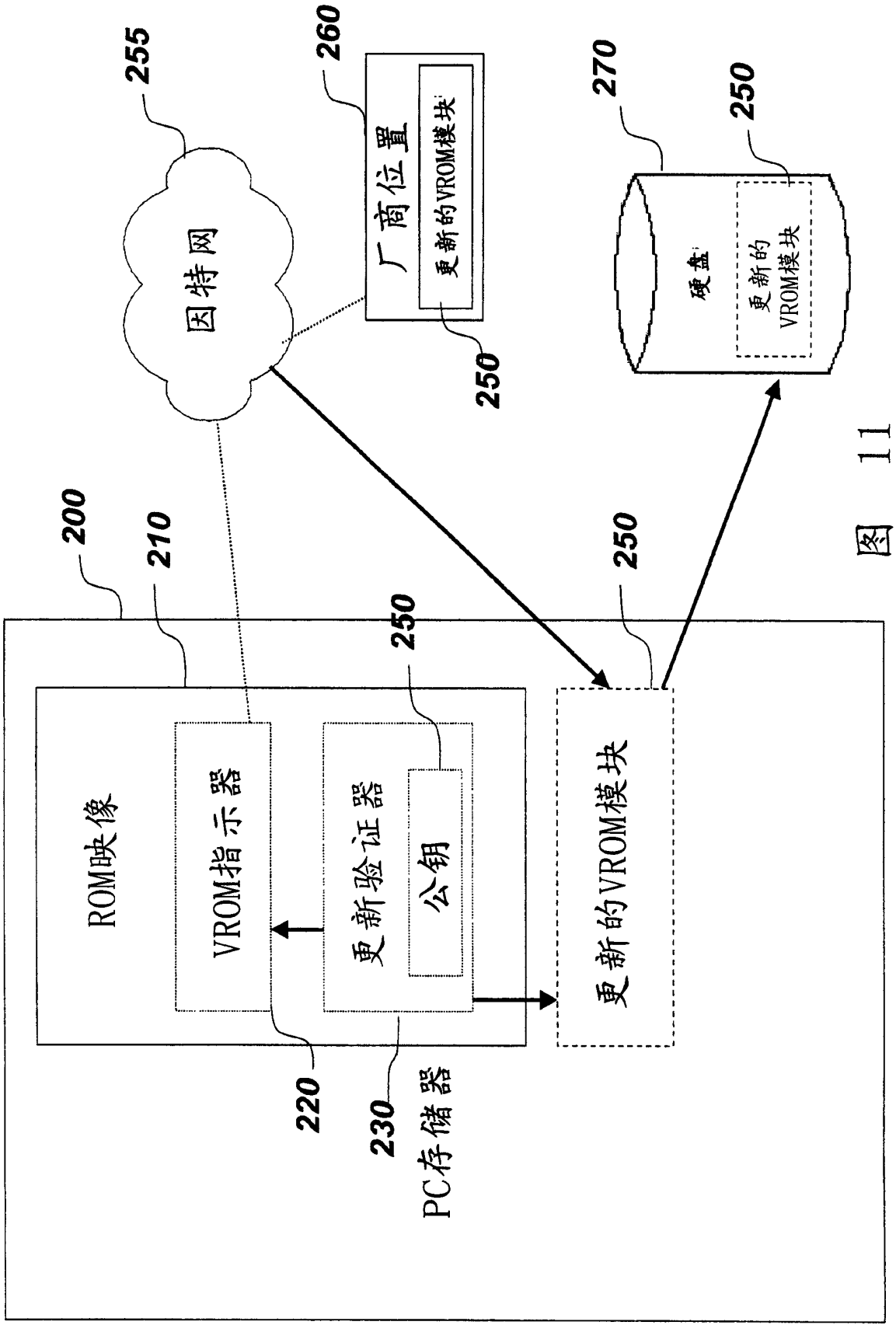

255
因特网
260
厂商位置
更新的VROM模块
250
270
250
硬盘
更新的
VROM模块
200
210
250
250
ROM映像
VROM指示器
更新验证器
公钥
更新的VROM模块
220
230
PC存储器

图 11

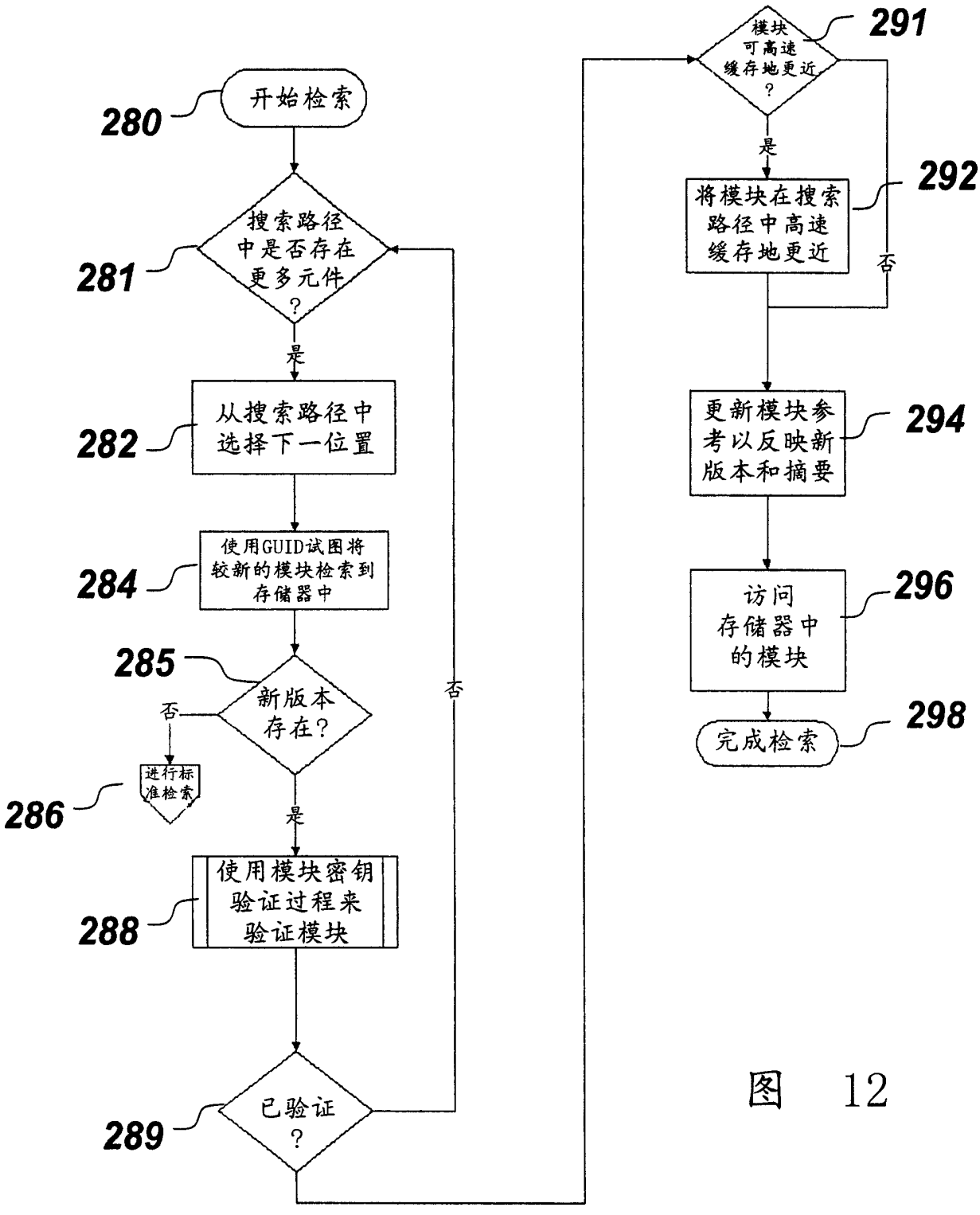

开始检索
280
搜索路径中是否存在更多元件？
281
是
从搜索路径中选择下一位置
282
使用GUID试图将较新的模块检索到存储器中
284
新版本存在？
285
否
进行标准检索
286
是
使用模块密钥验证过程来验证模块
288
已验证？
289
否
模块可高速缓存地更近？
291
是
将模块在搜索路径中高速缓存地更近
292
否
更新模块参考以反映新版本和摘要
294
访问存储器中的模块
296
完成检索
298

图 12

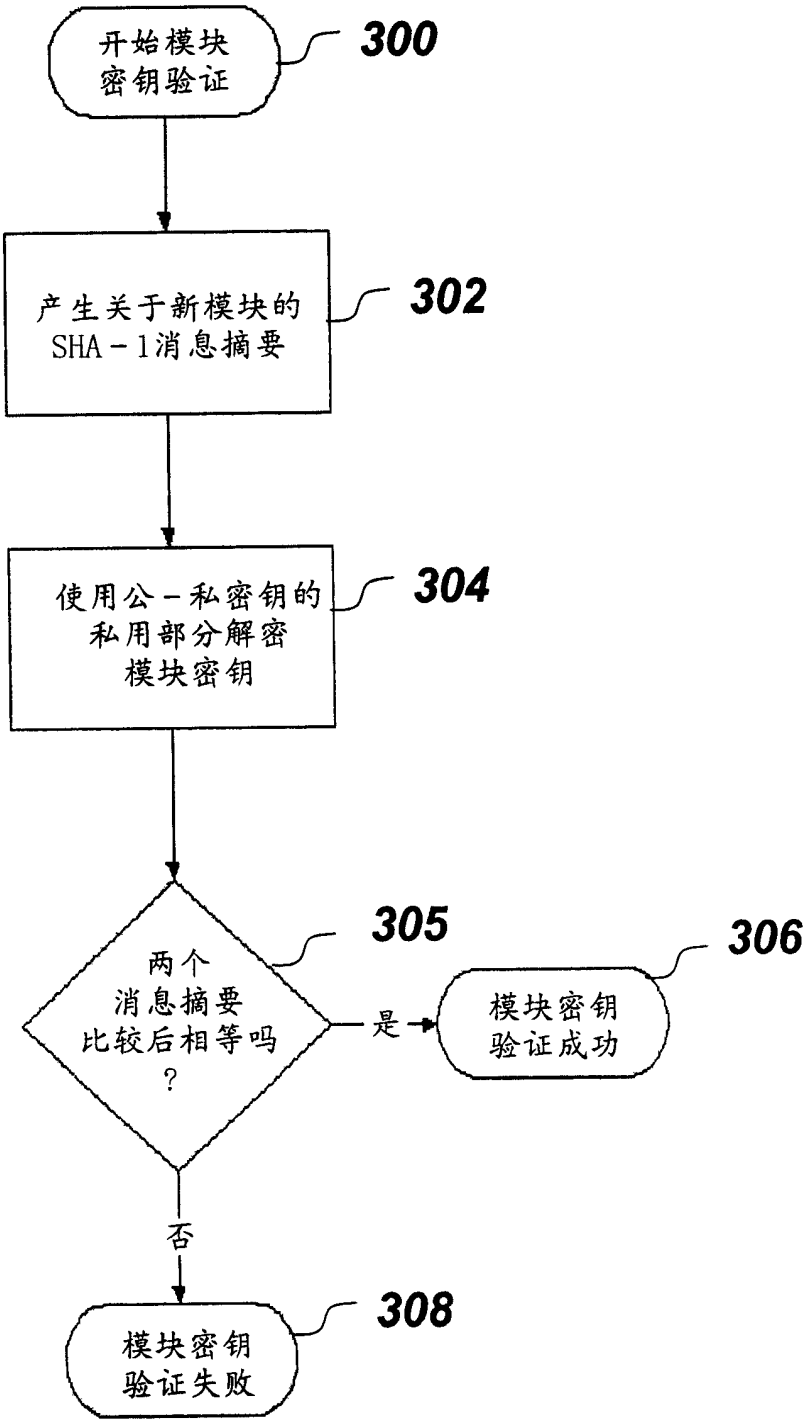

开始模块密钥验证
300
产生关于新模块的SHA－1消息摘要
302
使用公－私密钥的私用部分解密模块密钥
304
两个消息摘要比较后相等吗？
305
是
模块密钥验证成功
306
否
模块密钥验证失败
308

图 13

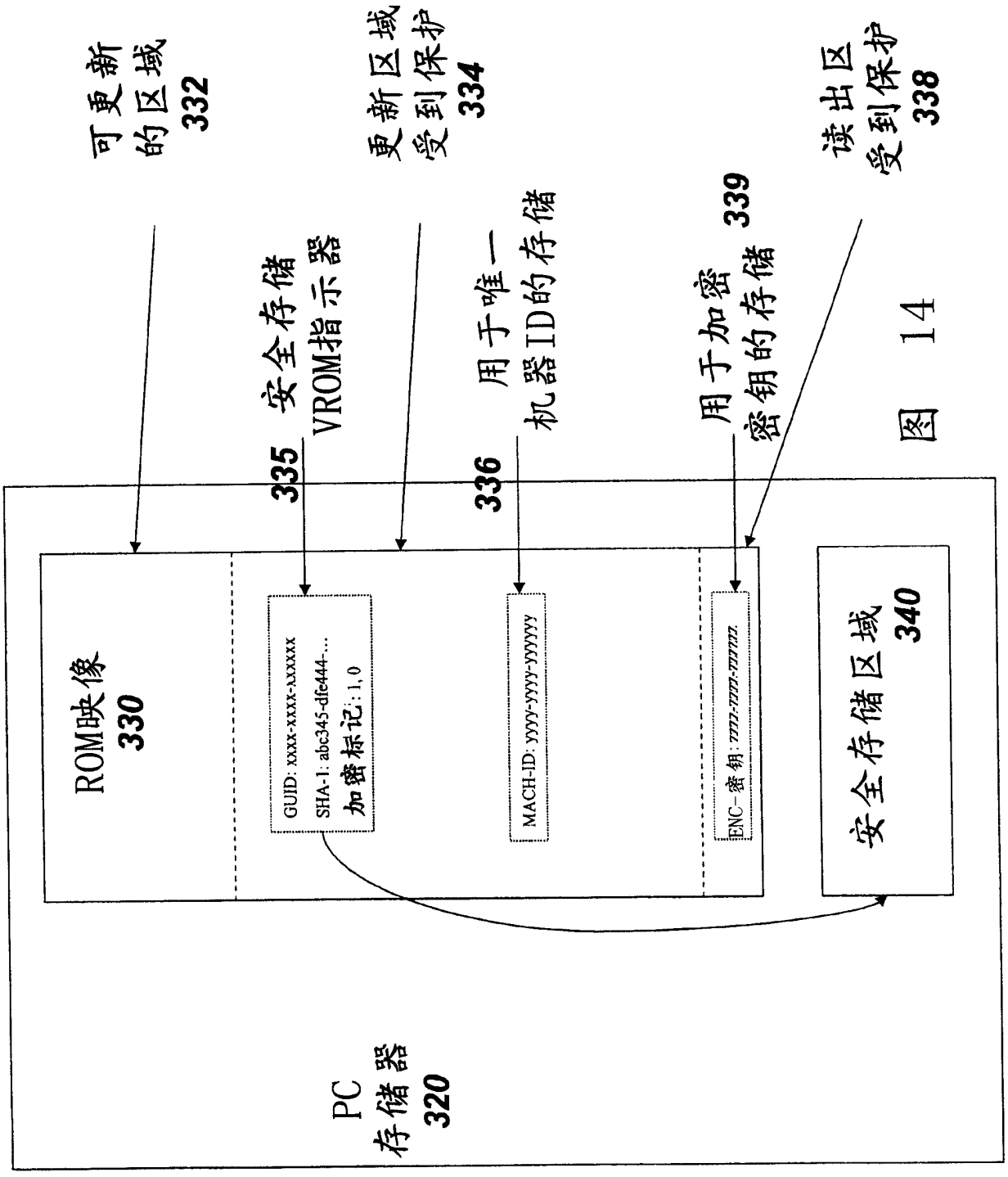

可更新的区域 332
更新区域受到保护 334
读出区受到保护 338
335 安全存储 VROM指示器
336 用于唯一机器ID的存储
用于加密密钥的存储 339
ROM映像 330
GUID: xxxx-xxxx-xxxxxx
SHA-1: abc345-dfe444-...
加密标记: 1, 0
MACH-ID: yyyy-yyyy-yyyyyy
ENC-密钥: ?????-?????-??????
安全存储区域 340
PC 存储器 320

图 14

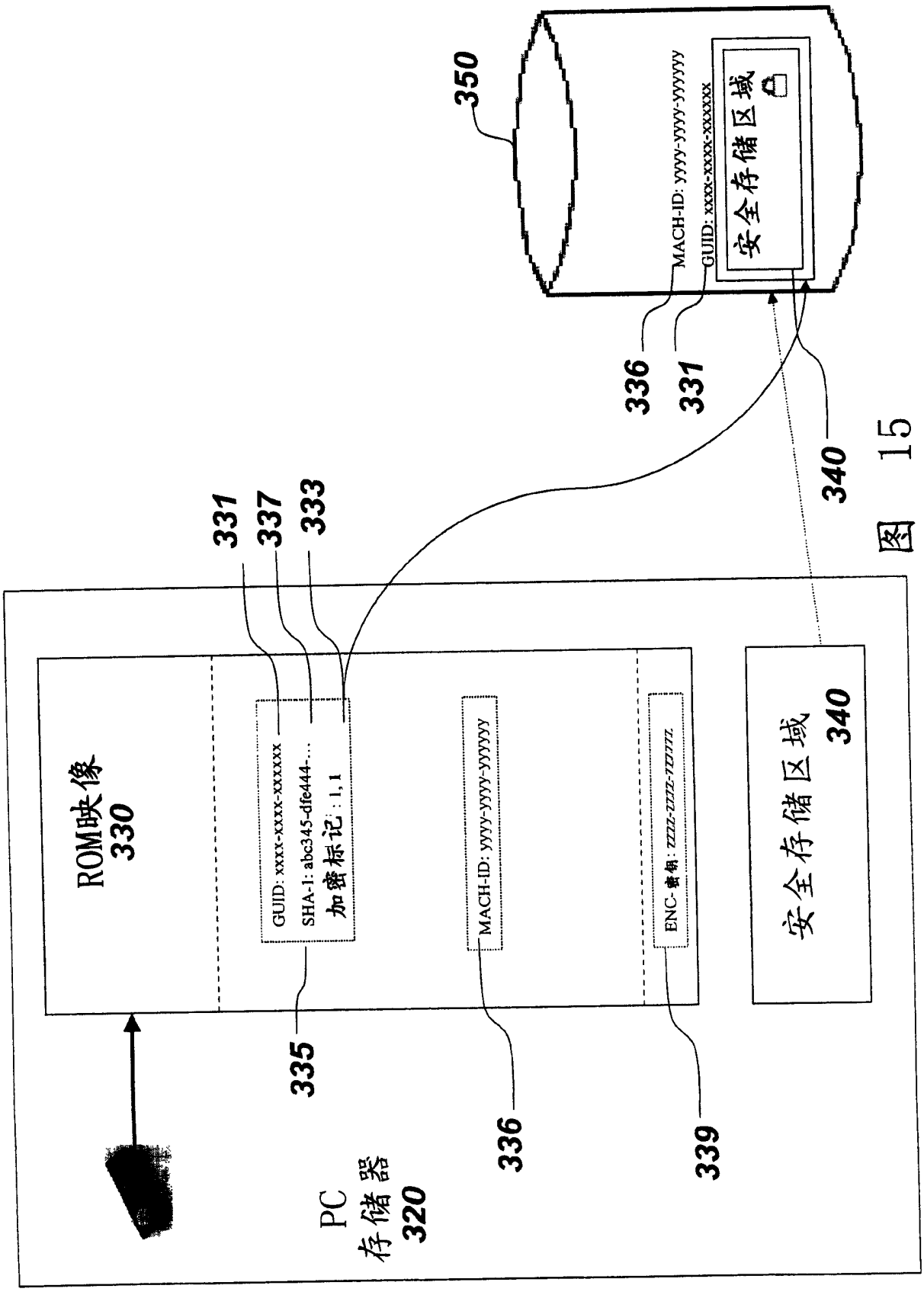

350
MACH-ID: yyyy-yyyy-yyyyyy
GUID: xxxx-xxxx-xxxxxx
安全存储区域
336
331
340
331
337
333
ROM映像
330
GUID: xxxx-xxxx-xxxxxx
SHA-1: abc345-dfe444-...
加密标记 : 1, 1
MACH-ID: yyyy-yyyy-yyyyyy
ENC-密钥: zzzz-zzzz-zzzzzz
安全存储区域
340
335
336
339
PC
存储器
320

图 15

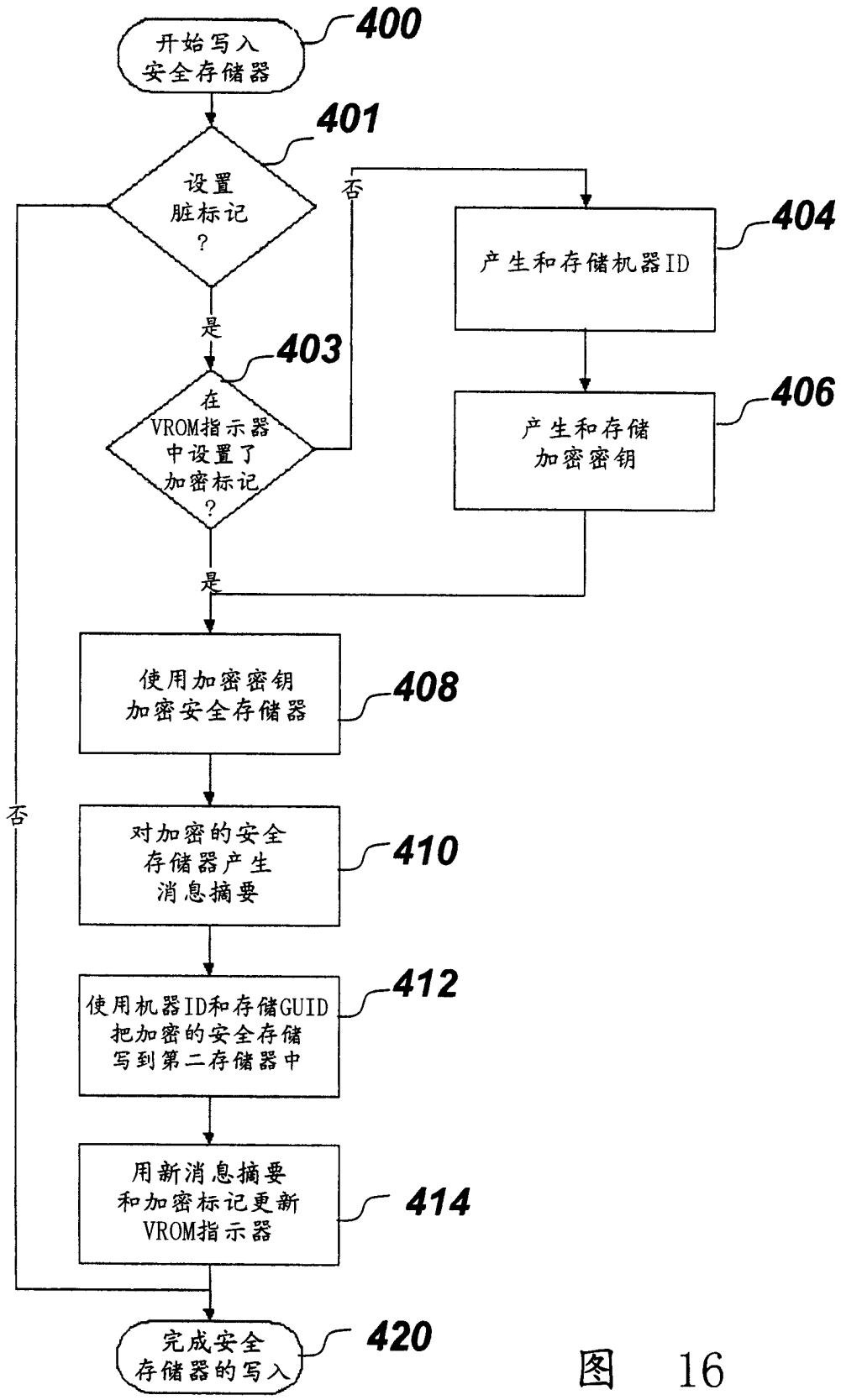

开始写入
安全存储器
400
设置
脏标记
?
401
否
是
在
VROM指示器
中设置了
加密标记
?
403
否
是
产生和存储机器ID
404
产生和存储
加密密钥
406
使用加密密钥
加密安全存储器
408
对加密的安全
存储器产生
消息摘要
410
使用机器ID和存储GUID
把加密的安全存储
写到第二存储器中
412
用新消息摘要
和加密标记更新
VROM指示器
414
完成安全
存储器的写入
420

图 16

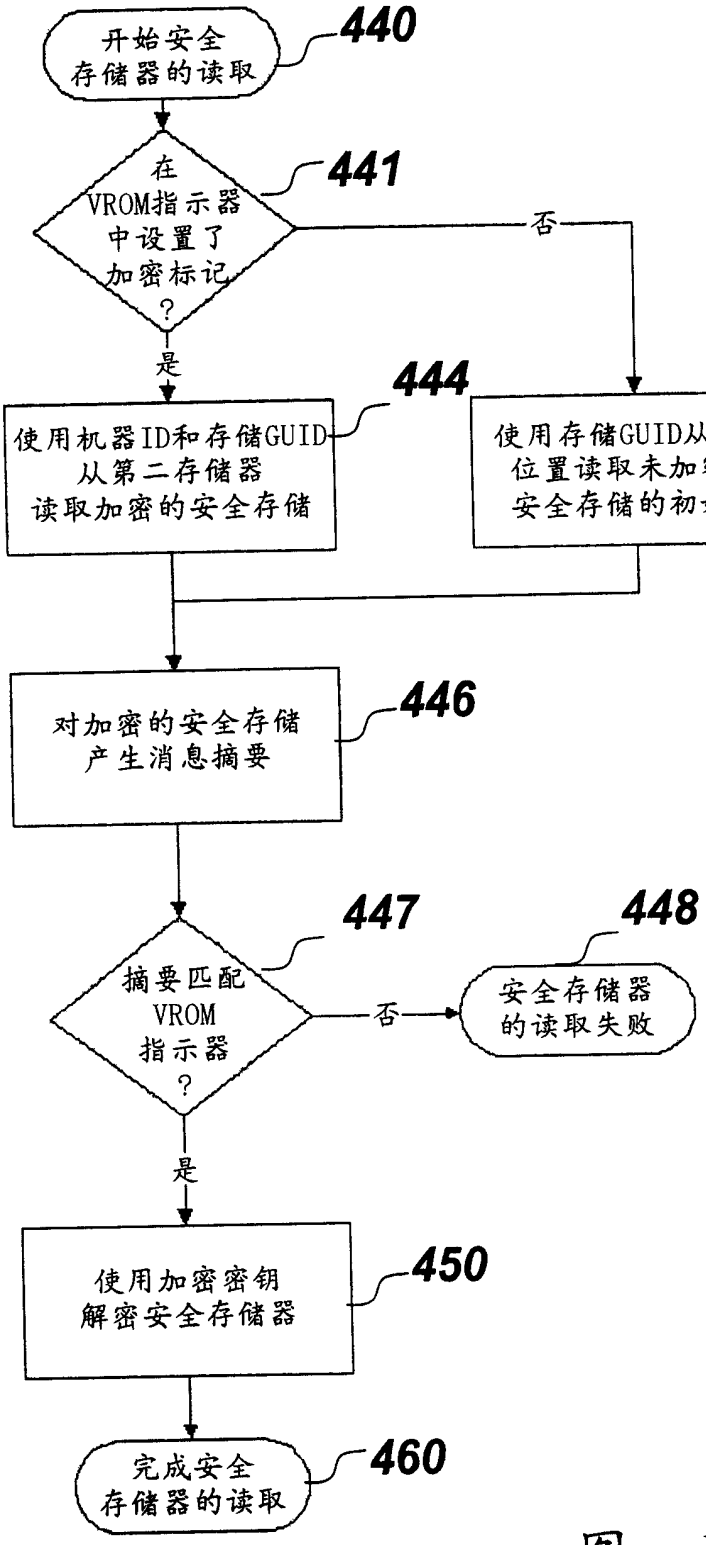

开始安全存储器的读取
440
在VROM指示器中设置了加密标记？
441
否
是
使用机器ID和存储GUID从第二存储器读取加密的安全存储
444
使用存储GUID从其它位置读取未加密的安全存储的初始值
442
对加密的安全存储产生消息摘要
446
摘要匹配VROM指示器？
447
否
安全存储器的读取失败
448
是
使用加密密钥解密安全存储器
450
完成安全存储器的读取
460

图 17